

## CHAPTER 1

# INTRODUCTION

### 1.1 Introduction

In today's digital age, education is increasingly moving towards interactive and engaging platforms, especially for young learners. The Kids Learning App project aims to create a mobile application that serves as a comprehensive educational tool tailored specifically for children aged 3 to 10 years old. This app will provide a fun and stimulating environment where kids can learn and explore various subjects through interactive games, videos, quizzes, and educational content designed to enhance their cognitive development and knowledge retention.

### 1.2 Problem Statement

Children's education is a critical foundation for their future success, but traditional learning methods may not always fully engage and motivate young learners. In today's digital age, there is a growing need for a comprehensive and interactive kids learning app that addresses the following challenges:

- **Lack of Engagement:** Many children find traditional educational materials boring, leading to a lack of interest in learning and limited knowledge retention. There is a need for an app that can capture children's attention, making learning a fun and enjoyable experience.
- **Personalized Learning:** Every child learns at their own pace and has different strengths and weaknesses. A kid's learning app should be able to adapt to individual learning styles and preferences, providing personalized content and challenges for each child.
- **Limited Accessibility:** Access to quality education can be limited in certain regions or communities. An effective kids learning app should be easily accessible on various devices, including smartphones and tablets, and offer offline capabilities to bridge the digital divide.
- **Safety and Security:** Online safety is a major concern for parents. The app must be designed with robust security measures to ensure a safe and child-friendly environment, free from harmful or inappropriate content.
- **Holistic Learning Approach:** Beyond academic subjects, children's holistic development is crucial. The app should incorporate interactive activities that promote cognitive, social, emotional, and motor skill development.

- **Parental Involvement:** Parents play a vital role in their child's education. The app should provide features that allow parents to track their child's progress, receive updates on learning milestones, and participate in their child's learning journey.

## 1.3 Objectives

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.
- 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
- Our project, "Learning Fun," is an engaging and interactive Android application designed to make learning enjoyable for kids. The app combines education and entertainment, offering a variety of games and activities that help children develop essential skills in reading, math, science, and creativity. Interactive lessons with animations and sounds teach basic concepts in different subjects, while exciting educational games reinforce learning through play. Creative activities like drawing, coloring, and puzzle-solving boost creativity and problem-solving skills. The app also features progress tracking, allowing parents to monitor their child's development and identify areas for improvement. With an ad-free and secure environment, "Learning Fun" ensures a safe learning space for kids, aiming to create a positive and enjoyable educational experience that helps children love learning and excel in their studies
- "Learning Fun" is designed with a child-centric approach, ensuring that every feature and activity is tailored to suit the learning styles and preferences of young children. The app's user interface is colorful and easy to navigate, encouraging children to explore and learn independently.
- The interactive lessons in "Learning Fun" are designed to make learning enjoyable. Each lesson is crafted to be visually appealing, with bright colors, fun animations, and engaging sounds. These elements help maintain the child's interest and make the learning process more effective. Lessons cover a wide range of subjects including:
  - Reading: Basic phonics, vocabulary, and comprehension exercises.
  - Math: Number recognition, addition, subtraction, and simple multiplication.

- The educational games are a key component of "Learning Fun." These games are designed to be both fun and educational, reinforcing the concepts learned in the lessons. Examples include:
  - Math Games: Counting games, addition and subtraction challenges, and shape recognition.
  - Reading Games: Word matching, phonics games, and spelling challenges.
  - Science Games: Simple experiments, nature exploration games, and quizzes on basic scientific facts.
- Each game is designed to be progressively challenging, ensuring that children remain engaged and motivated to learn.
- "Learning Fun" is more than just an educational app; it is a comprehensive learning platform designed to make education enjoyable and accessible for young children. By combining interactive lessons, educational games, creative activities, and robust progress tracking, the app provides a holistic learning experience that supports children's development in multiple areas. With its emphasis on safety and privacy, "Learning Fun" is a trusted companion for parents who want to give their children the best possible start in their educational journey.

## CHAPTER 2

# SOFTWARE REQUIREMENT SPECIFICATION

### 2.1 Functional Requirements

The functional requirements are the statement of services the system should provide, how system reacts to particular inputs and how system should behave in particular situation. It describes the functionality that the system provides. Our app requires:

- The user should have the appropriate version of windows.
- The application should be installed on the system.
- Active internet connection .

### 2.2 Non-functional Requirements

Defining and addressing the non-functional requirements (NFRs) for a system are among the most important of a software architect's responsibilities. NFRs are the system quality attributes for a system, as distinct from the functional requirements, which detail a system's business features and capabilities. Key points of nonfunctional requirements:

- **Performance and scalability:** How fast does the system return results? How much will this performance change with higher workloads?
- **Portability and compatibility:** Which hardware, operating systems, and browsers, along with their versions does the software run on? Does it conflict with other applications and processes within these environments?
- **Reliability, maintainability, availability:** How often does the system experience critical failures? How much time does it take to fix the issue when it arises? And how is user availability time compared to downtime?
- **Security:** How well are the system and its data protected against attacks?
- **Localization:** Is the system compatible with local specifics?
- **Usability:** How easy is it for a customer to use the system?

### 2.3 System Requirements

User requires active internet connection to use the app. User requirements are specific statements that describe what a user or a group of users need or expect from a product or system. These requirements outline the functionality, features, and characteristics that the product should possess to satisfy the users needs and provide value to them.

**Hardware Requirements:**

- Processor: intel/AMD processor.
- RAM: 8GB.
- Input: Keyboard/mouse.
- Display: Monitor.
- Memory: 4 GB

**Software Requirements:**

- Operating System: Windows 10
- Android SDK
- Android Studio

**Android SDK**

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. The ADT bundle includes the essential Android SDK components and a version of the Eclipse IDE with built-in Android Developer Tools to streamline the Android app development. ADT bundle consists of following components for developing the application II. Eclipse ADT plugin. The ADT bundle includes the essential Android SDK components and a version of the Eclipse IDE with built- in Android Developer Tools to streamline the Android app development. ADT bundle consists of following components for developing the application II.

**Android Studio**

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touch screen mobile devices such as smart phones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The current stable version is Android 12, released on June 5, 2023.

## CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

### 3.1 Existing System

The existing kids learning apps have made significant strides in providing interactive and engaging educational content for young learners. However, they are not without their drawbacks, which include the following:

#### Disadvantages

- **Limited Personalization:** While some apps attempt to offer personalized learning experiences, the level of customization may still be limited.
- **Overemphasis on Gamification:** Many kids learning apps heavily rely on gamification elements, such as rewards and badges, to motivate children.
- **Lack of Real-Life Application:** Some apps may present isolated educational content without sufficient connections to real-life scenarios.
- **Intrusive Advertisements:** Certain free apps might display advertisements, which can be distracting for young learners and may expose them to age-inappropriate content.
- **Limited Offline Access:** Not all kids learning apps offer offline access, which can be problematic for children in areas with limited internet connectivity or for parents concerned about excessive screen time.

### 3.2 Proposed System

The proposed kids learning app aims to overcome the limitations of the existing systems while providing a comprehensive, interactive, and personalized educational experience for young learners. Here are the key features and advantages of the proposed system:

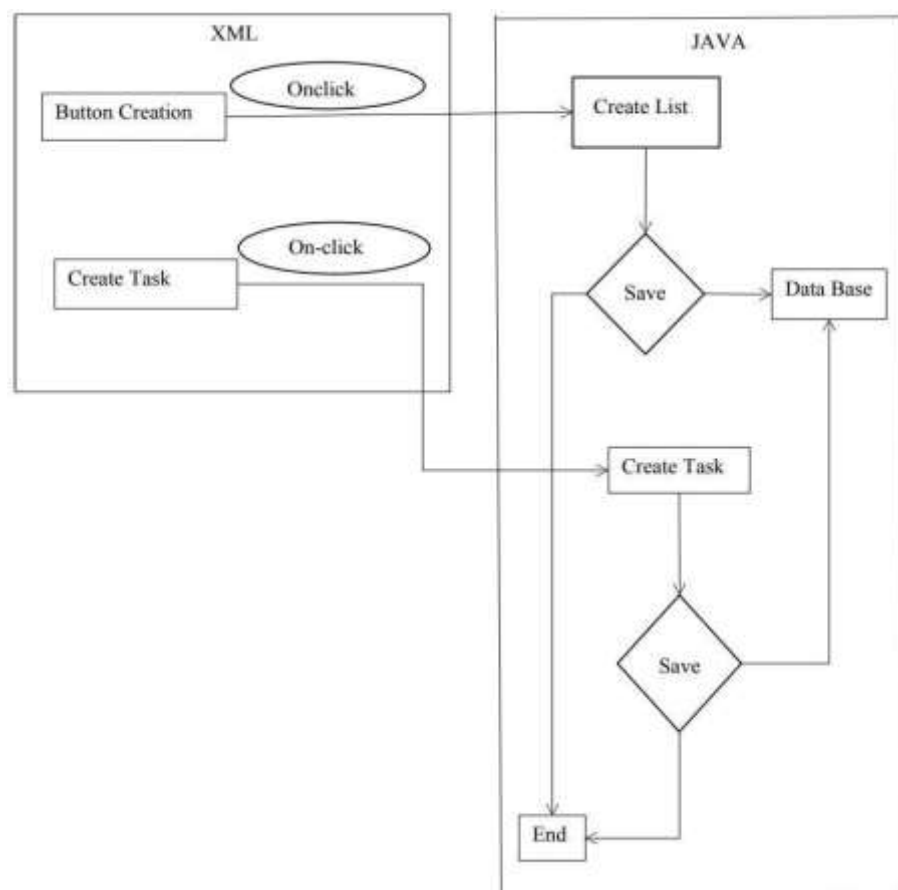
#### Advantages and Application

- **Personalized Learning Paths:** The app will utilize advanced AI algorithms to assess each child's strengths, weaknesses, and learning style.
- **Balanced Gamification:** The app will incorporate gamification elements thoughtfully, striking a balance between fun and educational value.
- **Real-Life Applications:** Educational content will be designed to have real-life applications, enabling children to see the practical relevance of their learning and encouraging critical thinking and problem-solving skills.

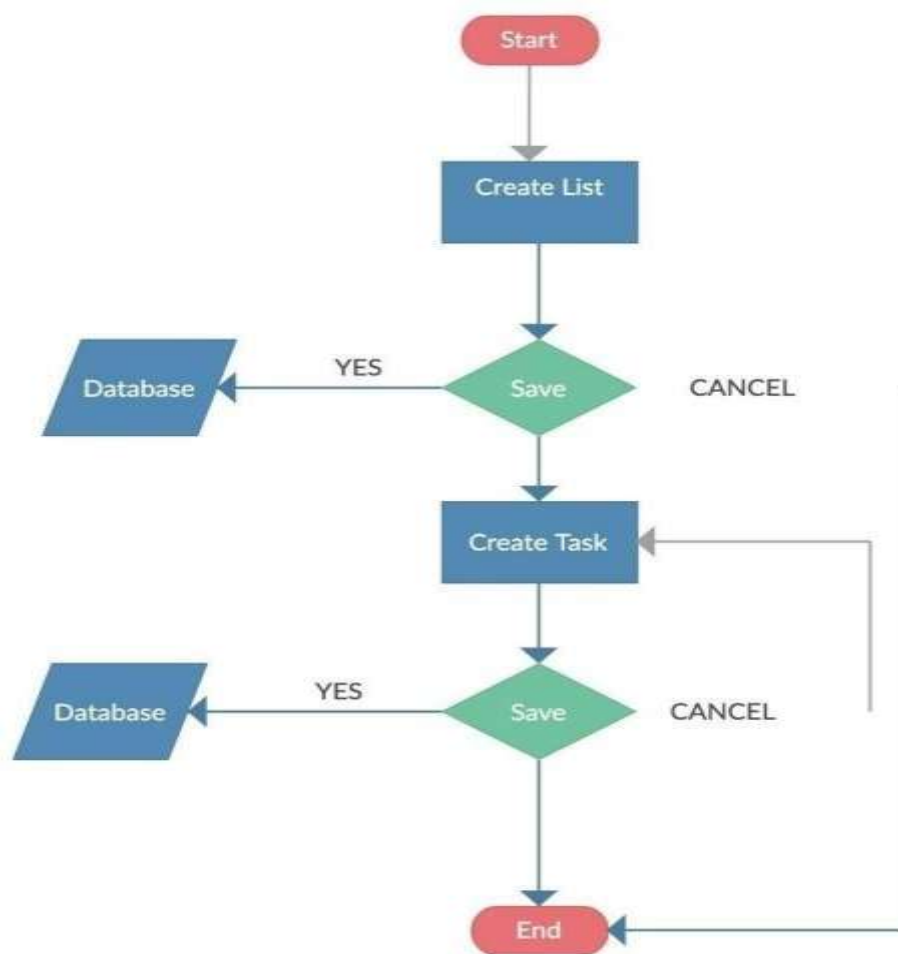
- **Ad-Free Experience:** The app will be completely ad-free, ensuring a distraction-free environment for children's learning and eliminating exposure to inappropriate content.
- **Offline Access and Low Data Usage:** To increase accessibility, the app will offer offline access to selected content, allowing children to learn even in areas with limited internet connectivity.

### 3.3 Architecture and Flowchart

The architecture of a kids learning app involves various components that work together to deliver a seamless and engaging educational experience for children.



**Fig 3.1 Architecture**

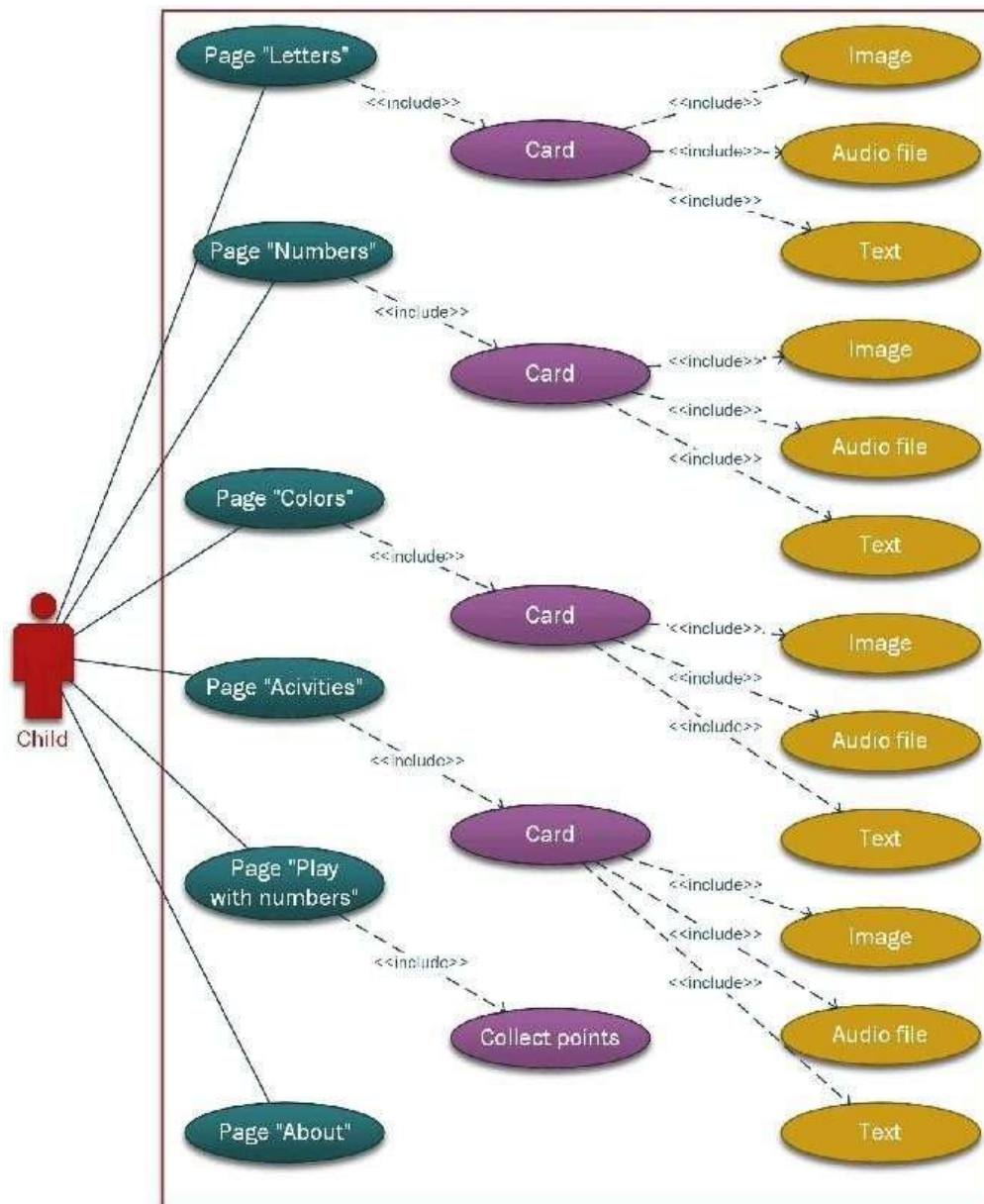


**Fig 3.2 Flow chart**

Creating a flowchart for a kids learning app would require a detailed understanding of the app's specific features and functionalities. Since I don't have information about the specific app you have in mind, I'll provide a general outline for a flowchart that could represent the navigation and interaction within a typical kids learning app.



### 3.3.3 Use Case diagram



**Fig 3.3 Use case diagram**

Use case diagram of game education application On application game education android-based English language page begins splash screen, later continued the home page, on this page the user can see the whole of the available services applied. In figure 2, we can see there are two actors that user and designer. Designers tasked to upload the application menu. Users are actors that interact directly with the application.

## CHAPTER 4

# IMPLEMENTATION

### 4.1 Pseudo Code

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools">
<RelativeLayout android:id="@+id/activity_homescreen"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingTop="5dp"
android:background="#ffffff">

<LinearLayout android:id="@+id/linetop"
android:layout_width="match_parent"
android:layout_height="wrap_content">

<ImageView android:layout_width="80dp"
android:layout_height="70dp"
android:background="@drawable/logo" />

<ImageView android:id="@+id/logout" android:layout_width="40dp"
android:layout_height="25dp"

android:layout_marginTop="5dp"

android:layout_marginRight="10dp"

android:src="@drawable/dots"

android:textColor="#ff0000"/>
```

```
</LinearLayout>

<ScrollView android:id="@+id/homeitems"

android:layout_width="match_parent"

android:layout_height="match_parent"

    android:layout_below="@+id/linetop"

android:layout_centerHorizontal="true">

<LinearLayout android:layout_width="match_parent"

android:layout_height="match_parent"

android:layout_gravity="center_horizontal"

android:orientation="vertical">

<LinearLayout android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center"

android:layout_marginBottom="15dp"

android:orientation="horizontal">
```

### **JAVA CODE**

```
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;
```

```
mpone = 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); mpsix=
MediaPlayer.create(AlphabetActivity.this, R.raw.f); mpseven=
MediaPlayer.create(AlphabetActivity.this, R.raw.g); mpeight=
MediaPlayer.create(AlphabetActivity.this, R.raw.h); mpnine=
MediaPlayer.create(AlphabetActivity.this, R.raw.i); mpten=
MediaPlayer.create(AlphabetActivity.this, R.raw.j); mpeleven =
MediaPlayer.create(AlphabetActivity.this, R.raw.k); mptwelve=
MediaPlayer.create(AlphabetActivity.this, R.raw.l); mpthrtten=
MediaPlayer.create(AlphabetActivity.this, R.raw.m); mpfoutteen=
MediaPlayer.create(AlphabetActivity.this, R.raw.n); mpfifthen=
MediaPlayer.create(AlphabetActivity.this, R.raw.o); mpsixten=
MediaPlayer.create(AlphabetActivity.this, R.raw.p); mpseventen=
MediaPlayer.create(AlphabetActivity.this, R.raw.q);

@Override

public void onViewLongClicked(int clickedViewId, int clickedItemPosition) {
}

@Override

public void onDestroy()
{
    super.onDestroy();
    mpone.release();
    mptwo.release();
    mpthree.release();
}
```

## CHAPTER 5

# TESTING

### 5.1 Testing

**Test Case:** A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards, guidelines and customer requirements. The process of writing a test case can also help reveal errors or defects within the system.

**Unit test cases.:** Unit testing involves analyzing individual units or components of the software to confirm each unit performs as expected. A unit is the smallest testable element of software. It often takes a few inputs to produce a single output.

**Integration test cases:** An integration test case is written to determine how the different software modules interact with each other. The main purpose of this test case is to confirm that the interfaces between different modules work correctly. Integration test cases are typically written by the testing team, with input provided by the development team.

**Functionality test cases:** This is a type of black box testing that can reveal if an app's interface works with the rest of the system and its users by identifying whether the functions that the software is expected to perform are a success or failure. Functionality test cases are based on system specifications or user stories, allowing tests to be performed without accessing the internal structures of the software. This test case is usually written by the QA team.

**Performance test cases:** These test cases can help validate response times and confirm the overall effectiveness of the system. Performance test cases include a very strict set of success criteria and can be used to understand how the system will operate in the real world. Performance test cases are typically written by the testing team, but they are often automated because one system can demand hundreds of thousands of performance tests.

**User interface test cases:** This type of test case can verify that specific element of the graphical user interface (GUI) look and perform as expected. UI test cases can reveal errors in elements that the user interacts with, such as grammar and spelling errors, broken links and cosmetic inconsistencies. UI tests often require cross-browser functionality to ensure an app performs consistently across different browsers. These test cases are usually written by the testing team with some help from the design team.

## 5.2 Test cases

Test Case ID	Test Case Description	Test Data	Expected Output	Actual Output	Test Status Pass/Fail
1	Click on colors button	Select the colors in the color page	Color names should be pronounced	Color names are pronounced	Pass
2	Click on numbers button	Select the numbers in the numbers page	Numbers should be pronounced	Numbers are pronounced	Pass
3	Click on alphabets button	Select the alphabets in the alphabets page	Alphabets should be pronounced	Alphabets are pronounced	Pass
4	Click on week days button	Select the days in the week days page	Days names should be pronounced	Days names are pronounced	Pass
5	Click on shapes button	Select the shapes in the shapes page	Shapes names should be pronounced	Shapes names are pronounced	Pass
6	Click on Poems/rhymes button	Select the poems in the poems page	Poems should be pronounced	Poems are pronounced	Pass
7	Click on rhymes button	Select the rhymes in the poems page	rhymes should be pronounced	Rhymes are pronounced	Pass

## CHAPTER 6

### RESULTS AND SNAPSHOTS

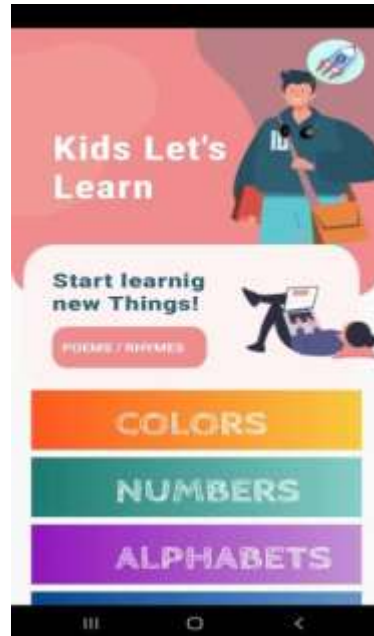


Fig 6.1 Home page

The home page displays the options available in kids learning app



Fig 6.2 Color page

The color page displays the color names to pronounce the selected color.



**Fig 6.3 Numbers page**

The numbers page will displays the numbers to pronounce the selected numbers.



**Fig 6.4 Alphabets page**

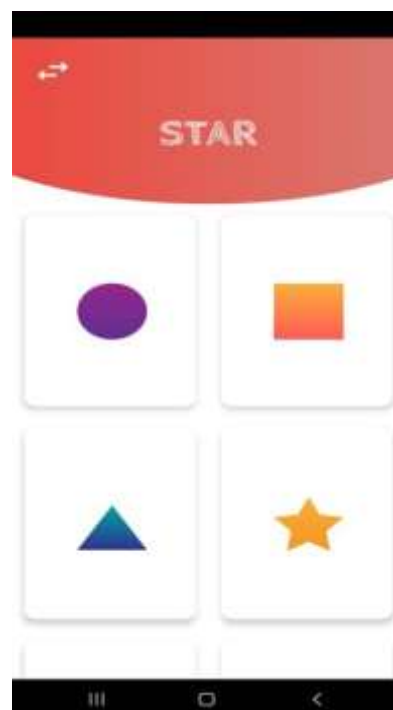
The alphabets page displays the alphabets names to pronounce the selected alphabets.





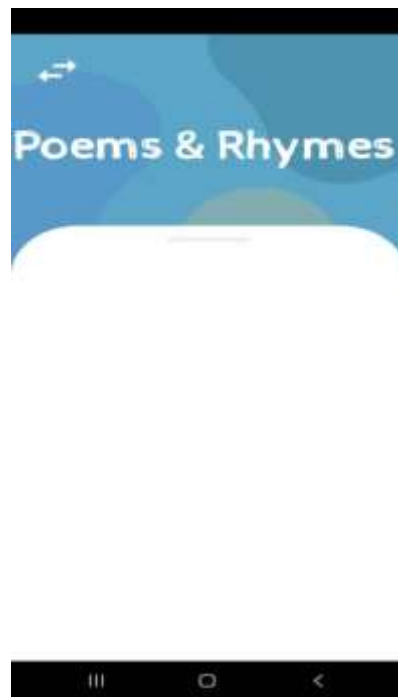
**Fig 6.5 Weeks page**

The week page displays the week names to pronounce the selected day.



**Fig 6.6 Shapes page**

The shapes page displays the shapes names to pronounce the selected shape.



**Fig 6.7 Poem page**

The poem page displays the poem to pronounce the selected poem.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **1. 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.

#### **2. 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 any real-life android project to this highly useful one.

## REFERENCES

- 1 Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017.
- 2 Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197.
- 3 Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPDPublishers, 2015. ISBN-13: 978-9352131341.
- 4 Bill Phillips, Chris Stewart and Kristin Marsicano, "Android Programming: The Big Nerd Ranch Guide", 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054.
- 5 Zao J.K., Mei-Ying Wang, Peihuan Tsai, Liu J.W.S., "Smart Phone Based Application".
- 6 Ashutosh Bhargave, Niranjana Jadhav, Apurva Joshi, Prachi Oke, S.R. Lahane, International Journal of Scientific and Research Publications 2013.
- 7 Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, "Implementing Customizable Web Based Application", International.
- 8 Kristin Marsicano, "Android Programming: The Big Nerd Ranch Guide", 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054.
- 9 S.R. Lahane, International Journal of Scientific and Research Publications.  
10 Priya Jadhav, Snehal Korade, Priyanka Teli, "Implementing Customizable Web Based Application"