

**MINI PROJECT - 02**  
**(2021-22)**

**“PHOTOEDITOR (Pico) APP”**

**Project Report**



**Institute of Engineering & Technology**

**Submitted By -**

Ritik Gupta (191500659)

**Under the Supervision Of**

**Ms. Ruchi Talwar**

**(Technical Trainer)**

**Department of Computer Engineering & Applications**



**Department of Computer Engineering and Applications**  
**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**  
**Chaumuha, Mathura – 281406 U.P (India)**

## **Declaration**

I hereby declare that the work which is being presented in the Bachelor of technology. Project “**PHOTOEDITOR (Pico) App**”, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of :-

**Ms. Ruchi Talwar (Technical Trainer) , Dept. of CEA , GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Sign : *RitikGupta***

**Name of Candidate:** Ritik Gupta

**University Roll No.:** 191500659



**Department of Computer Engineering and Applications**

**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**

**Chaumuha, Mathura – 281406 U.P (India)**

## **Certificate**

This is to certify that the project entitled “**PHOTOEDITOR (Pico) App**”, carried out in **Mini Project – II Lab**, is a bonafide work by **Ritik Gupta**, and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Signature of Supervisor:**

**Name of Supervisor:** Ms. Ruchi Talwar

**Date:**

## Technical Certificates

*Ritik Gupta*





**Department of Computer Engineering and Applications**  
**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**  
**Chamuha, Mathura – 281406 U.P (India)**

## **ACKNOWLEDGEMENT**

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Ms. Ruchi Talwar, my supervisor.

She has been helping us since Day 1 in this project. She provided us with the roadmap, the basic guidelines explaining on how to work on the project. She has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without her help, I wouldn't have been able to complete this project.

And at last but not the least I would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

**Sign:** *RitikGupta*

**Name of Candidate:** Ritik Gupta

**University Roll No:** 191500659

## **ABSTRACT**

In this project, we are creating an android application, basically a **PHOTOEDITOR App** which we have named **Pico**. This application will provide us a platform to access the photos we want to edit at the ease of our fingertips. All the users will be having their own photos on this app which will be save and share to various platforms like Facebook , WhatsApp , Instagram . Any photo that the user wishes to edit will be entered by him from the file/gallery folder which works on the basis of users choice input. The photo may be of any thing, which the image is related and will also work on specific keywords input. Apart from editing the photos , the user can save the photo he/she likes in the favourites folder. The app is suitable in the present scenario as the world is being digitalized then why not the photography system.. On the profile of the user, one can easily view the profile image he/she has upload. The app will be completely efficient and transparent to the reviews of the people on the camera quality and editing standard features. To get more details about the editing one can click on the image and get further editing features details. This app will be using The Google Photos API for providing all the photos. Further the Google Ads adds all the necessary details that the user may need about the app. The app also has a complete User Interface with all exciting editing features. There are some certain permissions that the user must allow at the starting of the application only once.

Android App ecosystem is diverse and is changing people's life all over the world. Android users are expected to increase because of the advance changes of the operating system and the way it deals with issues and compatibility with other mobile devices. Furthermore designing solutions for the problems that we may face in future is essential. Like this app definitely stands the need of the user at any time at their fingertips.

# CONTENTS

Cover Page.....	i
Declaration .....	ii
Certificate .....	iii
Technical Certificate.....	iv
Acknowledgement .....	v
Abstract .....	vi
Content .....	vii
List Of figures .....	ix
List Of tables .....	ix
Chapter 1 Introduction.....	1
• 1.1 Context.....	1 □
1.2 Motivation.....	1 □
1.3 Objective.....	1
• 1.4 Existing System.....	2 □
1.4 Sources.....	3
Chapter 2 Software Requirement Analysis.....	4
• 2.1 Impact Of Photos On Daily Life .....	4
• 2.2 Problem Statement.....	4
• 2.3 Hardware and Software Requirements.....	4-5

• 2.4 Modules and Functionalities.....	5
• 2.5 Photo editor Android Application.....,,,,,	6
Chapter 3 Technology Used.....	7
• 3.1 Android.....	8
• 3.2 Version of Android.....	8
• 3.3 Tools and Languages.....	9
• 3.4 Basic Terminology.....	10
Chapter 4 Implementation and User Interface.....	11
• 4.1 Implementation of Photo editor(Pico).....	11
• 4.2 User Interface.....	13-17
Chapter 5 Testing.....	17
• 5.1 Installation Testing.....	18
• 5.2 Unit Testing.....	18-23
• 5.3 User Testing.....	23
• 5.4 Performance Testing.....	23
• 5.5 Compatibility Testing.....	24
Chapter 6 Conclusion.....	24
<b>References.....</b>	<b>25</b>



## LIST OF FIGURES

1. Existing System.....	2
2. Android Kit-Kat.....	7
3. Splash Screen.....	13
4. Google Admob screen.....	13
5. Navigating and selecting image.....	14
6. Editing page (a).....	14
7. Editing page (b).....	15
8. Editing page (c).....	15
9. Saving Page.....	16
10. Sharing Page.....	16
11. Interstitial Page.....	17

## LIST OF TABLES

1. Version of Android.....	8
2. Unit testing of Photo Editor(Pico).....	18-23

# CHAPTER-1

## INTRODUCTION

### 1.1 CONTEXT

This Android Application “**PHOTOEDITOR (Pico)**” has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by **Ms. Ruchi Talwar**. This project has been completed approximately three weeks and has been executed in modules, meetings have been organized to check the progress of the work and for instructions and guidelines.

### 1.2 MOTIVATION

In the recent years, we have realized the importance of selfies/photos posted on social media platforms and how important it is for the youth and other people to have our resources online. Photos have been the greatest source of interest all the while and having editing them at the reach of our fingertips would be an opportunity hardly of any user would afford to miss.

In the century we are living the world is progressing at a really great pace, a lot number of technologies come up every single day. To keep up with the technology is also important to survive in this world of digitalization. Along with this we need to have a place to keep the resources for areas of our interest so we thought of developing an app which could provide us with many photos as well as a platform where we could edit the photos as we like to edit to make it more attractive. Adding to its features, some of the photos will edit up into the editor box according to the recent search, making you go deep into the field you are editing in.

Moreover this kind of application can be used in photography where any age user can make changes to their photos. This would be an excellent effort to provide photography without any boundaries to all.

### 1.3 OBJECTIVE

The main objective of this application is to create a PHOTOEDITOR app named “Pico” which will have a lot of images and a space to keep up the image one wants to edit. There will be a facility to search any image one wishes to edit by the use of any standard tool like the filters, frames, round, exposure, contrast, vignette, saturation, warmth, pixelate, draw, sticker and text.

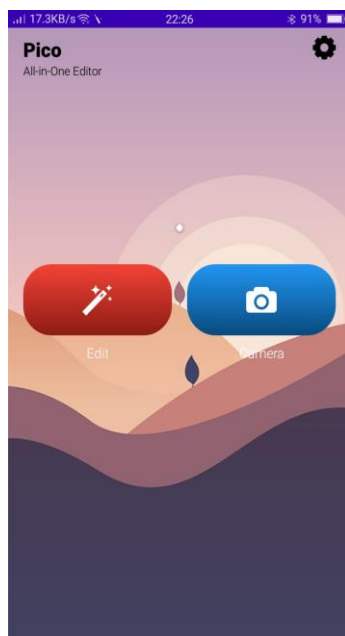
After the editing there will be list of related share/upload options like Facebook, WhatsApp and Instagram and one can view and like more about the details of the image and can further share it.

This application developed can be used at a variety of places, at photography studios and have its significance. The goal of the app was to provide a way to the users to get all the images in the way they want they desire to edit at a particular image location rather than randomly surfing on the Internet.

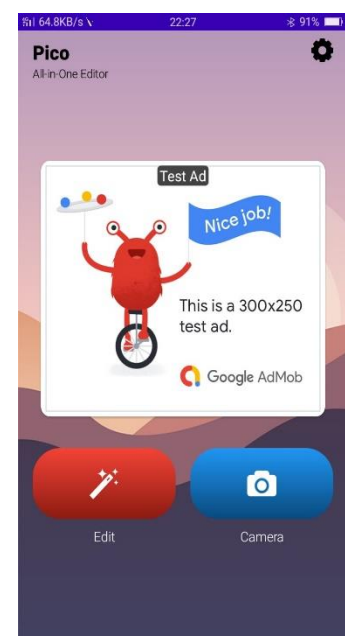
## 1.4 EXISTING SYSTEM

In the present scenario, we are dealing with the manual searching of photo editor from thousands of other photo editors present in Google Play store API. With the help of this application we are able to find a place where we can easily edit or caption the image with the help of standard tools. As this idea as already implemented here are the some snap how our application will look.

As soon as the user enters the app, there will be landing page containing the Home page of the app and then there will be a setting icon also with a google ads in that page .Initially there will be buttons clickable as shown in the image below. Then on the basis of certain option like camera and edit the app will fetch the images captured or use for editing as shown in the second screenshot. In this we will add a feature to select the image from various folders in the mobile this will be helpful when the user try the use the image again.



(a)



(b)

**Fig 1- Existing Systems**

## **1.5 SOURCES**

The source of our project (including all the project work, documentations and presentations) will be available at the following link

<https://github.com/ritikgupta2109/PHOTOEDITOR--Pico>

## **CHAPTER -2 SOFTWARE REQUIREMENT ANALYSIS**

### **2.1 IMPACT OF PHOTOS ON DAILY LIFE**

Photos are considered as the best thing of all people in a real sense, and it is said that they are also the best companions of to say without words. They play a significant role in People's life. Photos give plenty of joy to users as the memory, and they learn a lot of things from photos. They take them into a unique world of imagination and improve their standard of living.

Photos help to inspire users to do nice work with innovation. They enrich the experience of users and sharpen their innovative thinking skills.

Images processing increase the knowledge of users and improve their mindsets. They reveal the different concepts and introduce the numerous shades of culture of the world. Clicking images makes users aware of the various societies and civilizations across the globe. By clicking photos ones can explore the past, present and future and can view many things. Photos inspect the clarity and creativity in user's mind . Photos can help improve the memory of users. When users see photos, they create the images of the story and character mentioned in that image and they

memorize what they have saw. So if users click/edit regularly their visual memory will activate and it will help them to quickly remember things. .

It naturally help them to focus on their life. This will also help them to improve their interest and they can learn more in very less time. Clicking/editing can enhance the imagination and creativity of user's and increases positive thinking.

## **2.2 PROBLEM STATEMENT**

The Photo Editor App “Pico” is an Android Application which will allow the users to search any image using photos name into which the image folder falls in or any other keyword related to the photo. Actually this app is connected to the Google Admob and every time the user gets an ad for the image , the image is searched on the API details are reverted back to the user's screen.

Along the side, for the users a free folder space is being provided for them to store the images they like in the “Favourites Section”. As a help to the users, there will be sharing image section containing the basic details and upload on various platforms. The profile of each user is created and can be updated anytime as per the requirements. One another feature that our app holds is as soon as a various standard operations on image can be performed , it is put up on the API and since we have directly connected to the API it can also be found on our app without any modifications by the creators.

This app is a complete image editing and caption with all the facilities a user desires and with the proper User Interface as well.

## **2.3 HARDWARE AND SOFTWARE REQUIREMENTS**

### **Hardware Requirement**

- Processor :intel i5
- Operating System : Any Operating System
- RAM : 8 GB (or higher)
- Hard disk : 256GB

## **Software Requirement**

- Software used: Android Studio
- Language used : Android JAVA
- Others : Google Admob
- User Interface Design : Android Application

## **2.4 MODULES AND FUNCTIONALITIES**

- **Splash Screen:** The first screen with which the user interacts will be this screen containing the logo and the app name .This will disappear within 5 seconds after the app is displayed.
- **Navigation Drawer:** This is the most important part of the application that provides interactivity within the app as it connects the various activities together like it is a side bar on which the dashboard, the favourites section, the About page of the page are linked and on clicking on each you can visit the pages.
- **Dashboard Page:** This is the page displayed for every user after entering the app successfully. It contains the edit icon where the user can search the photo to be edit according to the wish as well as some of the photos are suggested with the genres recently searched or the most popular one.
- **Favourites Page:** Initially the page is empty, but when the users search for an image and like it then one can add it to the favourites section. This place is a user's personal space to store any image he likes or want to edit.

## **2.5 PHOTOEDITOR (Pico) ON ANDROID APPLICATION**

*“Pico”* is actually a image editing platform. This is the place with the collection of large number of edited images. Some platforms will be having a collection of millions of images so their arrangement and maintenance is also important. But it will be very difficult for people search for a particular image among millions and hence it comes up as tedious job. Even the experienced

editors may fail to find a particular image and its location so here is when an android application “Pico” comes into picture. *This app will give users new ways to show their love for images share them online, keep track of what they share of their imageshelf with friends, and, of course, discover new photos to edit.*

## **CHAPTER-3 TECHNOLOGY USED**

### **3.1 ANDROID**

Android is a linux-based operating system designed primarily for touch screen devices such as smart phone tablets and computers. Released in 2008, is now owned by Google. So android is a operating system like Windows, Ubuntu and Mac OS and a lot number of devices use Android these days like mobile phones, watches, laptop and television. So we also created an android application “Pico”, a library of photos. Play Store is a market place for all the Android Apps. So we need to know what basically an android app is. An Android app is software running on a Android Platform. So this can be concluded that like all the software it is a combination of Backend and Frontend. Backend to design the logical parts of the app, for the functionality whereas Front End to develop the User Interface. And to implement the various parts of the android app, we require a number of tools and technologies which will come into picture. But first it would be great to see the three different type of Android Apps:-

- **Native Apps:** An executable program coded in the machine language of the hardware platform it is running in. **Native applications** are compiled into the machine language of that CPU. For example, **Windows** and Mac executable **apps** are in x86 machine language, while **mobile apps** are ARM based. Native apps are the most common. They're coded in a specific language like Swift for **iOS** or Java for Android. A popular example is WhatsApp.
- **Web Apps:** are accessed via the internet browser and will adapt to whichever device you're viewing them on. They are not native to a particular system, and don't need to be

downloaded or installed. Due to their responsive nature, they do indeed look and function a lot like mobile apps — and this is where the confusion arises.

- **Hybrid Apps:** Hybrid apps are deployed in a native container that uses a mobile Web View object. When the app is used, this object displays web content thanks to the use of web technologies (CSS, JavaScript, HTML, HTML5). It is in fact displaying web pages from a desktop website that are adapted to a Web View display. The web content can either be displayed as soon as the app is opened or for certain parts of the app only i.e. for the purchase funnel. In order to access a device's hardware features (accelerometer, camera, contacts...) for which the native apps are installed, it is possible to include native elements of each platform's user interfaces (iOS, Android): native code will be used to access the specific features in order to create a seamless user experience. Hybrid apps can also rely on platforms that offer JavaScript

APIs if those functionalities are called within a Web View

## 4.2 VERSION OF ANDROID

Each year Android releases a new version with better features, better security and better User Interface experience and a new symbol. Here is the table of list of versions



**Figure-5: Android Kitkat**



Code name	Version number	Initial release date
(No codename)	1.0	September 23, 2008
Petit Four	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Eclair	2.0 - 2.1	October 26, 2009
Froyo	2.2 - 2.2.3	May 20, 2010
Gingerbread	2.3 - 2.3.7	December 6, 2010
Honeycomb	3.0 - 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	October 18, 2011
Jelly Bean	4.1 - 4.3.1	July 9, 2012
KitKat	4.4 - 4.4.4	October 31, 2013
Lollipop	5.0 - 5.1.1	November 12, 2014
Marshmallow	6.0 - 6.0.1	October 5, 2015
Nougat	7.0 - 7.1.2	August 22, 2016
Oreo	8.0 - 8.1	August 21, 2017
Pie	9.0	August 6, 2018

**Table -1: Versions of Android**

## 4.3 TOOLS AND LANGUAGES

Tools used to build the Android App are:-

- **Android Studio:** Android Studio is an environment that help us create and edit Android applications. It is the official IDE for Android App Development. It has intelliJ's powerful code editor and developer tools and various features that enhance productivity while developing apps.
- **Software Development Kit (SDK):** Android Studio requires a collection of libraries and data therefore SDK is mandatory.

Languages used in building an Android Application are classified as per the Front End and Back End. For designing the Front End of an application we have used JAVA and Google Admob and for designing the Back End only JAVA.

⇒ **JAVA :-** Java is easy to learn for a variety of reasons. There's certainly no shortage of Java resources out there to help you learn the language, including websites, tutorials, books, and classes. Java is one of the most widely discussed, taught, and used programming languages on the planet. It's used for many different types of programming projects, no matter their scale, from web applications to desktop applications to mobile applications.

## 4.4 BASIC TERMINOLOGY

- **Layout**: Layout is the parent of view. It arranges all the views in a proper manner on the screen.
- **Activity**: An activity can be referred as your device's screen which you see. User can place UI elements in any order in the created window of user's choice.
- **View**: A view is an UI which occupies rectangular area on the screen to draw and handle user events.
- **Emulator**: An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.
- **Manifest file**: Manifest file acts as a metadata for every application. This file contains all the essential information about the application like app icon, app name, launcher activity, and required permissions etc.
- **API**: Short for Application Programming Interface. APIs are functions that developers can call on to access specific features by calling upon programs, code, and services that others have written. For example, if a developer wants to draw a button on the screen, she can insert a small bit of code that says “draw this kind of button, with this color and size and style, at this location” instead of dozens of lines of code that tells the graphics processor, in detail, exactly how to draw a button. If the application wants your location, it can use the location API to “get the device's location” and let Google's code handle the rest, instead of requiring the developer to build an entire location service from scratch just for her own app. There are thousands of APIs in Android, covering everything from drawing interface elements, to the cameras, to location access, to accessing storage, to 3D graphics (see: OpenGL ES) and much more.
- **Intent**: Intents are an essential part of the Android ecosystem. They are used to express an action to be performed. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. It can be classified into implicit and explicit intents.
- **Implicit intent**: It does not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.

- **Explicit Intent**: It specifies the component to start by name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.
- **APK**: Short for "Android application package." The extension used in Android app installation files (e.g., app.apk). Similar in nature to an EXE file on Windows.
- **SDK**: Short for "Software Development Kit." As it pertains to Android, the SDK is a set of tools such as code libraries, a debugger, and a handset emulator that can be run on Windows, Mac, or Linux to facilitate the creation of Android apps by developers. While the SDK is generally intended for use by developers, end users can install the software on their home computer to execute ADB and Fast boot commands.
- **Action Bar**: The action bar is an important design element, usually at the top of each screen in an app that provides a consistent familiar look between Android apps. It is used to provide better user interaction and experience by supporting easy navigation through tabs and drop-down lists.
- **Navigation bar**: Android Navigation Drawer is a sliding left menu that is used to display the important links in the application. Navigation drawer makes it easy to navigate to and fro between those links. It's not visible by default and it needs to be opened either by sliding from left or clicking its icon in the Action Bar.
- **Fragment**: A Fragment represents a behavior or a portion of user interface in a Fragment Activity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities.
- **JSON** stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and retrieving) from the server. Hence knowing the syntax and its usability is important.

JSON is the best alternative for XML and its more readable by human

## **CHAPTER -4**

### **IMPLEMENTATION AND USER INTERFACE**

Creating an app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making the concept clear to the developer is probably the most important factor in successful mobile app development. Yet it is one of the most common problems or obstacles in a mobile app development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

#### **4.1 Implementation of the Pico:**

Implementation of Pico is taken place in various phases. Firstly we build the dashboard interface then Navigation drawer i.e. make fragment for each of the list item using the Navigation view and the make various layout for the supporting features and connect the app with the Google API and Google Admob for fetch the required photo using camera and edit buttons. And finally we parse the Json object to get the data in the required format and then display the result.

##### **4.1.1 Step to be followed to develop the app:**

1. Firstly we create the splash screen with animated text using XML and linked it with the main Activity through JAVA.
2. After that we create dashboard phase which comprises of various phases that are mentioned below:
  - Setting icon: allows user to make changes into the app if the user is existing one .
  - Permission pop-up : If the user is new to our app then firstly he/she have to allow to use various things on the app such as camera and image file location.
  - Camera icon : To capture the photo.
  - Edit icon : To edit the image using various standard operations.

- Google Admob: To show some interstitial ads and reference ads.
3. Now, we are going to create Navigation drawer for that purpose we have used following functionality of android:
- Fragments(SupportFramentManager)
  - Menu – items
  - Drawer header
  - Hamburger icon
  - ActionBarDrawableToggle (help to create navigation Bar)
4. .Creating fragment for each of the menu item. Our Menu items are:
- Dashboard
  - Camera
  - Google Admob
  - Edit
  - Favourites
  - Save &Sharing
5. Now we have created various activities like Photos List on the device and many more.

In this step we connect our app with the GOOGLE ADMOB API using Lollipop version Android apps easier and most importantly, faster .

i.After that we parse the JSON object that we have received as a response for our query to get the data in the standard form.

ii. Now we add data (that we have received from Google API) to the photo editing activity .

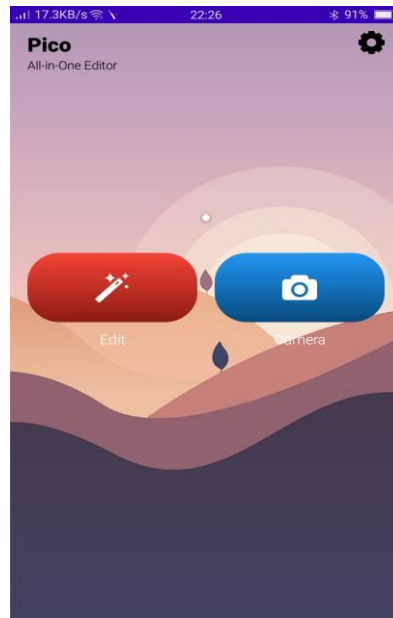
iii. In the editing Activity there are various functionality. Some of them are mentioned below

a. Preview: We have set OnClickListener to this. With the help of this if user clicks on this button it will redirect the user to the Google Admob related page on Google.

b. Favourites : it will add the photos to the favourites that you can view to later.

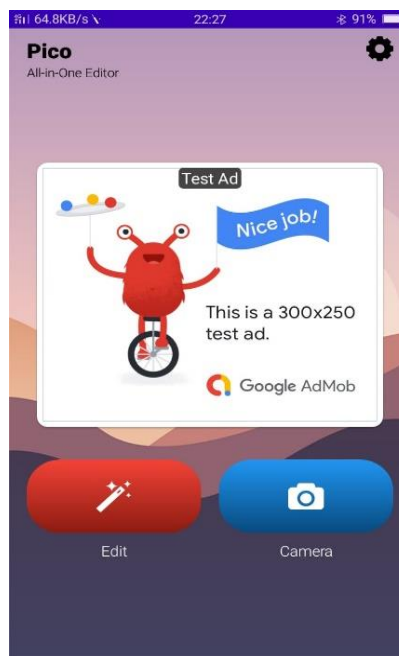
## 4.2 User Interface

- **Splash Screen**



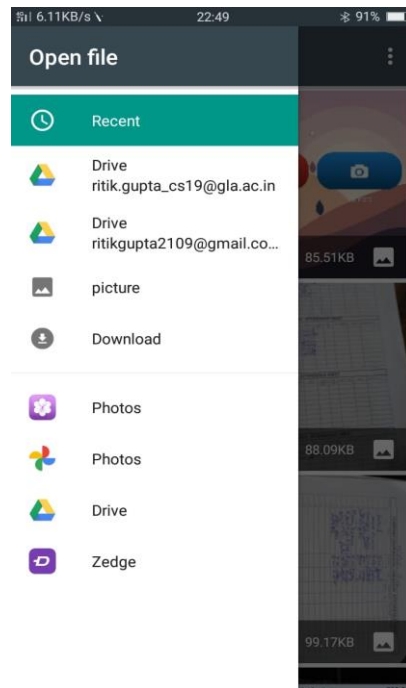
**Fig-2: Splash Screen**

- **Dashboard with Google Admob screen**



**Fig-3: Google Admob**

- **Navigating and selecting images**

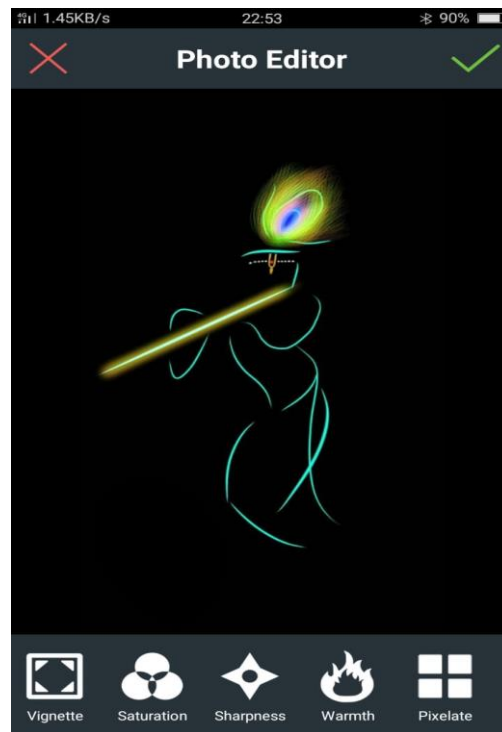


**Fig-4: Navigating & selecting image**

- **Editing screens**



**Fig-5: Editing Page(a)**



**Fig-6: Editing Page (b)**



**Fig-6: Editing Page (c)**

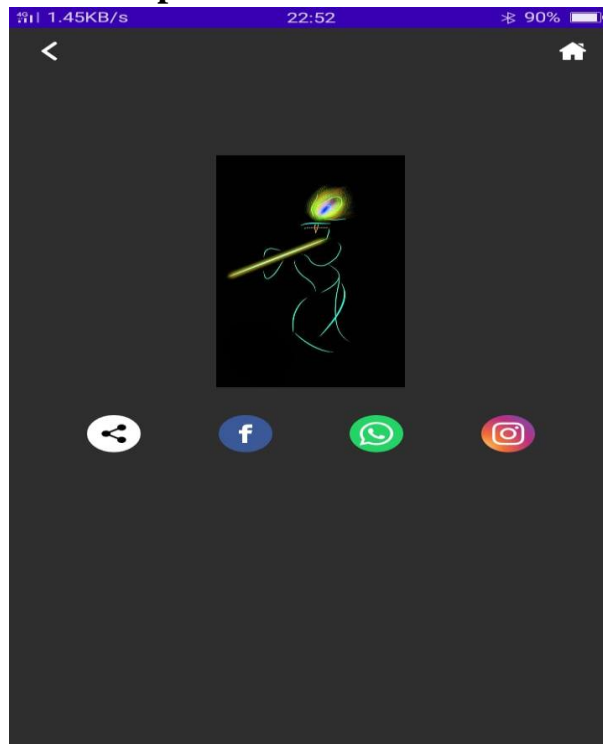


- **Saving image**



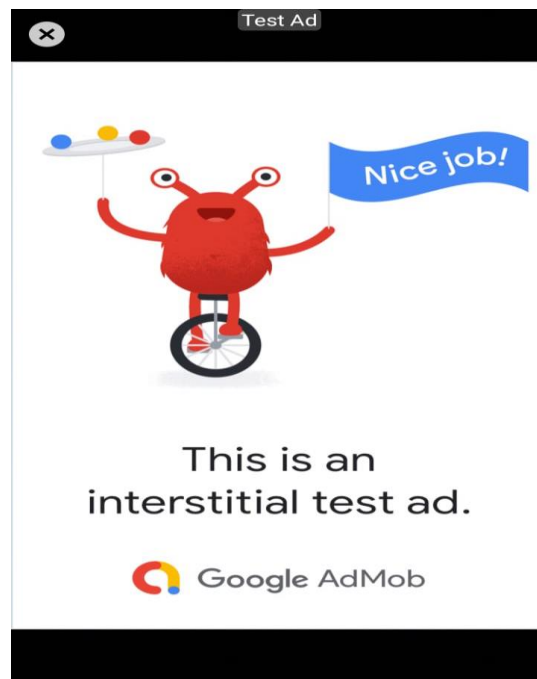
**Fig-7: Saving image**

- **Sharing image on various platforms**



**Fig-8: Sharing image on various platforms**

- **Interstitial Ads**



**Fig-9: Showing Interstitial ads**

## **CHAPTER - 5**

### **TESTING**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques.

System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data. In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find

whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting.

There are different types of testing some of them are listed below:

### **5.1 Installation Testing:**

There are two types of apps on an Android device i.e., Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing is carried out by our teammates. It is ensuring smooth installation of the application without ending up in errors, partial installation etc.

### **5.2 Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs. In this testing technique we are primarily focuses on

- Loop methods and function is working fine or not. □ Misunderstood or incorrect Arithmetic precedence
- Incorrect Initialization

#### **Unit Testing of the app:**

<b>Test cases</b>	<b>Description</b>	<b>Expected Outcome</b>	<b>Result</b>

1	Start Page – Launch Screen	Should display splash screen with animated setting icon	Pass
2	Dashboard with google admob screen	Should display google admob activity where you find edit and camera button option	Pass

3	Navigating and selecting image	Should display the dashboard of selecting an image from the device	Pass
4	Editing screens	Editing Page – (a) Filter, Frame, Round, Exposure, Contrast.	Pass

5	Editing screens	Editing Page – (b) Vignette, Saturation, Sharpness, Warmth, Pixelate.	Pass
6	Editing screens	Editing Page – (c) Drawer, Sticker, Text.	Pass

7	Saving image	Should saved image in the device After editing	Pass
8	Sharing image on various platforms	Should share edited image on platforms like – Facebook, WhatsApp, Instagram.	Pass

9	Interstitial Testing Ads	Should display testing ads from Google Admob	Pass
---	--------------------------	--	------

**Table 1: Unit Testing of Photo Editor (Pico)**

### **5.3 User Testing**

User testing is the process through which the interface and functions of a website, app, product, or service are tested by real users who perform specific tasks in realistic conditions. The purpose of this process is to evaluate the usability of that website or app and to decide whether the product is ready to be launched for real users.

This app was tested by me and friends who are using different mobile phones (and having different android version) also tested on different emulator to check its performance and it seems to be working fine and users of this app are satisfied with the facilities and performance of the app and like the way how the app is worked.

### **5.4 Performance Testing**

In this type of testing we have checked the performances of our application under some peculiar conditions are checked. Those conditions include:

- Low memory in the device.
- The battery in extremely at a low level.
- Poor/Bad network reception.



Performance is basically tested from 2 ends, application end, and the application server end. Our app is also performing well in this phase of testing as well. And we are getting positive feedback from user of our app.

### **5.5 Compability Testing**

This application was tested and used on different devices like LG G3, Google Nexus 4, Pixel 21. The application worked fine and is stable. The application worked fine in portrait mode and there isn't any problem with compatibility.

On all types of testing (that we have performed above) our performing well on our app i.e. Photo Editor (Pico).

## **CHAPTER -6**

### **CONCLUSION**

Proposed Pico App is an android application that will allow users to search for photos to edit. This application takes in a user input and searches the Google Photos and Google Admob API with the user input and gets a list of images on the device based on the users search query. Search result screen will contain a list of image with following details: After editing user can save or share on social – media platforms . Users can also add the photos to the favourites folders.

This application has wide range of scope in the upcoming era. It is impossible to arrange the hard copies of every book so this type of application can reduce the barrier to get knowledge at any place in a cost effective, productive way. For students who are interested in learning online can use this application and keep all the books they want to learn from at one place (in favourites section) and can create their own personal library of photos. Even individual image stores can have this system of image editing apps promoting their brand name as Digital Image Processing and can gain number of users.

## REFERENCES

**1. Introduction to Android:**

<http://developer.android.com/guide/index.html>.

**2. *Android API:***

<http://developer.android.com/reference/packages.html>

**3. *Android User Interfaces:***

<http://developer.android.com/guide/topics/ui/index.html>

**4. *Layout:***

[http://developer.android.com/guide/topics/ui/declaring-  
layout.html](http://developer.android.com/guide/topics/ui/declaring-layout.html)

**5. *Android Training:***

<http://developer.android.com/training/index.html>.

**6. Android developer Guide: <https://developer.android.com/>**

**7. For rectifying the error :**

<https://stackoverflow.com/>