

A
PROJECT
ON

“PHOTOSHOP AND EDITING APPLICATION”

Submitted to

Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)

In the Partial Fulfillment of

B.Com. (Computer Application) Final Year

Submitted by
Krutika Bansod
Palak Yadav

Under the Guidance of

Pravin J. Yadao



Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)

2022-2023

Shiksha Mandal's
**G. S. COLLEGE OF COMMERCE & ECONOMICS,
NAGPUR
(AUTONOMOUS)
CERTIFICATE**

(2022 - 2023)

This is to certify that Mr. /Miss Krutika Bansod and Palak Yadav has completed their project on the topic of Photoshop And Editing Application prescribed by G. S. College of Commerce & Economics, Nagpur (Autonomous) for B.Com. (Computer Application) – Semester-VI.

Date:

Place: Nagpur

Pravin J. Yadao

Project Guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and whole hearted thanks to project guide Prof. Pravin Yadao, Coordinator for his guidance throughout this work. We are very much thankful to him for his constant encouragement, support and kindness.

We are also grateful to our teachers Prof. Rahul Tiwari, Prof. Sushma Gawande, Prof. Preeti Rangari, Prof. Prajкта Deshpande and Prof. Haresh Naringe for their encouragement, help and support from time to time.

We also wish to express our sincere thanks to Principal Dr. S. S. Kathaley for providing us wide range of opportunities, facilities and inspiration to gather professional knowledge and material without which this project could not have been completed.

Student Names & Signature

Date:

Place: Nagpur

DECLARATION

We **Miss. Krutika Bansod and Palak Yadav** hereby honestly declare that the work entitled “**PHOTOSHOP AND EDITING APPLICATION**” submitted by us at G. S. College of Commerce & Economics, Nagpur (Autonomous) in partial fulfillment of requirement for the award of B.Com. (Computer Application) degree by Rashtrasant Tukadoji Maharaj, Nagpur University, Nagpur has not been submitted elsewhere for the award of any degree, during the academic session 2022-2023.

The project has been developed and completed by us independently under the supervision of the subject teacher and project guide.

Student Name & Signature

Krutika bansod

Palak Yadav

Date:

Place: Nagpur

INDEX

Sr. No	Particulars	Page No	Sign	Remark
1.	INTRODUCTION	6-11		
2.	OBJECTIVES	12-14		
3.	PRELIMINARY SYSTEM ANALYSIS 3.1 Preliminary Investigation 3.2 Present System In Use 3.3 Flaws In Present System 3.4 Need For New System 3.5 Feasibility Study 3.6 Project Category	15-26		
4.	SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATION	27-28		
5.	DETAILED SYSTEM ANALYSIS 5.1 Data Flow Diagram 5.2 Numbers of Modules and Process Logic 5.3 Data Structures and Tables	29-33		
6.	SYSTEM DESIGN 6.1 Form Design 6.2 Source Code 6.3 Input Screen and Output Screen	34-75		
7.	TESTING AND VALIDATION CHECKS	76-86		
8.	SYSTEM SECURITY MEASURES	87-89		
9.	IMPLEMENTATION,EVALUTION MAINTENANCE	90-97		
10.	FUTURE SCOPE OF THE PROJECT	98-99		
11.	CONCLUSION	100-101		
12.	BIBLIOGRAPHY & REFERENCES	102-103		

INTRODUCTION

The success of the present system depends largely on the how accurately the problem is defined, theory investigated and properly carried out through the choice of the solution. My project application is based on editing of Images. Photoshop and Editing Application is a basic graphics utility. In this application we introduce the heart of the digital process: Image Editing the art of turning raw images.

Original Image



Edited Image for Crop



Image editing can be used to Draw, Colors, Crop, Filter, Adjust, Mirror, Rotate, Edit and different Effects. One can fill the multiple colors, effects, etc. it has great versatility and is useful for both beginners and professional graphic. Image editor gives access to user for various controls.

This is a runtime program and also, you can save your image. Hence, the user can simply use this app to kill their time with joy. This interesting project is in the python programming language. Talking about the features of this Application, this python application is designed to remove the borderless of any person.

This project consists of many exciting features like crop, filter, adjust, mirror, rotate, effects, colours, etc. You can save your edited pictures and you can click on the restart to start from the beginning.

Also, the User Interface of this Application is pretty simple so that the user won't get any difficulties while working on it. So over all Image editor is the simple raster graphics tool. The principal features of image editor are **simple editing tools** that you can use to easily paint, effects on a blank canvas or existing image.

Image editor is an application that offers very little in the way of advanced features. It is **really easy to use**, even for user will not require any tips or manual for how to operate the same. The uncomplicated user interface featured tools i.e, menus appear on the top.

The image produced by scanners and digital cameras are often quite good, but rarely perfect. They may suffice as records of a scene or event, but they have the dramatic impact of a great print. If your goal is to make prints that go beyond simple records to capture the essence people and places to create prints that stands as works of art that you will need to edit the image. Image editing is the heart of the creative act of photographic printmaking it is where you transform a well crafted snapshot into a work of art. An image editor application with various image editing functionality that allows you to crop, zoom, transform, adjust, brightness, and apply more such transformations over an image.

Edited Image for different filters

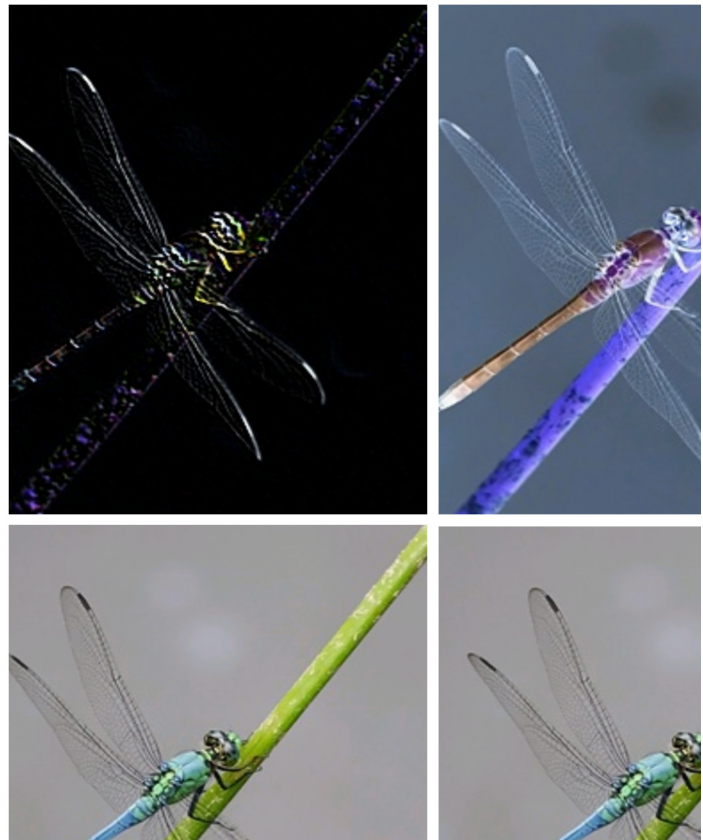


Image editing encompasses the processes of altering images, whether they are digital photographs, traditional photo-chemical photographs, or illustrations. Image editing can be pretty daunting. Open up any post-processing program, and you will be confronted by a slew of sliders, tools, and options.

Fortunately, basic photo editing is not nearly hard as it seems. There are few simple steps you can take to get great results.

Specifically, we are offering you a series of basic principles that you can use when editing any image of yours. They are quick, easy, and they will remain useful, even as you level up your editing skills with help of our photo editing application. Even a large expensive camera does not guarantee high-quality images. Technique alone is not always enough. That is why editing apps come into slightly correct the photo, add some details, give it a different character or style.

Original Image



Edited Image for mirror



Image editing encompasses the processes of altering images, whether they are digital photographs, traditional photo-chemical photographs, or illustrations. Image editing can be pretty daunting. Open up any post-processing program, and you will be confronted by a slew of sliders, tools, and options. Fortunately, basic photo editing is not nearly hard as it seems. There are few simple steps you can take to get great results.

Specifically, we are offering you a series of basic principles that you can use when editing any image of yours. They are quick, easy, and they will remain useful, even as you level up your editing skills with help of our photo editing application. Even a large expensive camera does not guarantee high-quality images. Technique alone is not always enough. That is why editing apps come into slightly correct the photo, add some details, give it a different character or style.

Here are some features of our application explained below:

This application system first allows the user to choose an image from your local device. A menu bar on the top of the screen provides this functionality. The menu bar allows the user to open a new image and even save an edited image in your device. The application allows you to zoom in and out of the image. User can even apply texts (captions) on images. It allows the user to crop and rotate the image as needed. Application also allows user to add reddish, bluish and greenish effect to the image. A user can also save the edited image into the camera roll. Finally user may save the edited file as needed.

Contrast simply refers to the difference between the darks and lights in a photo, where intense darks and intense lights creates high contrast, and softer darks and softer lights creates low contrast. Though you should always be careful not to take the contrast too far; otherwise, you'll end up creating a garish, nightmarish effect. By the way, if you're after a softer, more ethereal look, you may want to decrease the Contrast slider. That's what photographers often do when shooting in foggy conditions, where a low-contrast look enhances the mood.

Saturation refers to the intensity of the colors in your image, which you can adjust you guessed it the Saturation slider:

By boosting the saturation, you can add pop to your image, and so I do often recommend you add a bit of saturation but carefully, like seasoning to a meal. It's easy to go too far, and then you'll end up with an unpleasant result.

Some post-processing programs offer an alternative, called vibrancies. This is like saturation, but tends to be more subtle. If saturation isn't giving you the look you're after, go ahead and try vibrancies instead. Or start by trying vibrancies, then move on to saturation. The order isn't important; it's all about experimenting to achieve the effect you want.

On occasion, you should even decrease the vibrancies and/or saturation. For instance, if you want a moody image but the colors are just too bright and powerful, try reducing the saturation. Here, a little change can go a long way, and by subtly dropping the saturation, you can get an interesting cinematic effect.

OBJECTIVES

Let's start with defining what a photo editing app is. Well its name speaks for itself: It's mobile or desktop software that helps a user edit the images and save them in various formats.

Its core consists of two main parts:

- **Pixel processing:** This implies working with pixels settings like colour, brightness, reflections, contrast.
- **Layers processing:** With the help of new layers, you can add filters, decorations, or even another image.

The typical flow of working with photo editors looks like this: You upload a photo, edit its parameters with tools and present, choose a filter if needed, save a picture and share it on social media.

There are plenty of photo editing apps that work like that. So it's really necessary to create another one? The answer is yes, as we will explain.

- **Reasons to build a photo editing app:**

1. The over-beautification of images is no longer an ideal, but the photo editing specialization is still in demand. The market continues to grow.
2. Each day brings new, so its very possible that your product will capture the spotlight.
3. Photo editing can be a way of getting a second chance to get the image the way you would have wanted to capture it in camera.
4. Editing is also another way you can put your own personal stamp on your image.
5. Certainly, its not just editing that allows you to do that, but it is another way of making your way of viewing the world stand out, or have a certain feel that makes the image feel like "you".

6. Photo editing is another way for you to tell your story through it, or get across what you were trying to capture in the camera sometimes does not accurately reflect that moment.
7. Firstly, because we might not get our settings quite right in camera, but also because you can enhance a mood or a general feeling in processing that you will never really be able to fully capture in camera.
8. There are benefits associated with image editing. It enhances the original images in accordance to the users requirement. They can bring more colour and life to the image. It helps in bringing the best possible image in the interests of the viewers.
9. Image editing refers to modifying or improving digital or traditional photographic images using different techniques, tools or software.
10. Images produced by scanners, digital cameras or other image-capturing devices may be good, but not perfect. Image editing is done to create the best possible look for the images and also to improve the overall quality of the image according to different parameters.

PRELIMINARY SYSTEM ANALYSIS

All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a requirement specification document is created.

Each and every application has a need to be developed in such a way that the information which is to be given must be easily understandable and presentable. Every system has a common Things and that are. It is organized combination of different components. They are independent and inter-related. They work for common objectives. Thus in order to carry out project successfully it should be analyzed properly purposes of preliminary system analysis is to find that whatever the project will be successful or not. The project will be possible with all available resources like cost, time, human resources current business environment and technology. The System analysis is the method where the whole system is studied very in detail and as the basis the analysis of a complete product is presented in front of the users. The phases which should be studied in the preliminary system are as-

- Identification of Need
- Preliminary Investigation
- Feasibility Study
- Need of new system.
- Flaws in present system

Identification of Need

Before working on any system, management must analysis the need of prospective users. System should be according to the requirements of users. Management must know what the problem is before so that same can be solved. We can find out the real problem of the existing system. With clear understanding of the problem, we can take lessons make it advantage of current system. So it is necessary to identify the actual problem. In order to carry out any project successfully the first and foremost requirement is to identify the need to carry out the project. The system analyst should first of all derive the needs or the objective of the project. This helps him to make analyst that whether the particular system that is currently undertaken is feasible or not. The project **IMAGE EDITOR** is opted for promoting the joyful image editing application. In this application, we have created a menu

in order to let the use any kind of draw, effects, adjust frame, etc onto the screen using the controls provided on the menu.

Preliminary Investigation

Preliminary investigation is the first step. In this step, the system is investigated. The objective of this step is to conduct an initial analysis and findings of the system.

While searching for an idea we had only one think in our mind that it must be useful for the user. So we come up with an idea to build an application with different types of image editing application so it may explore user's mind. It may help user not only in their present but also for future.

This is the stage where the is being carried out by the developer who actually wants to develop a project. Any person related with the system initiate the request. This problem is investigated in the preliminary investigation. This is the stage where the need and requirement of the user identified being investigated and develop a new system as per user requirements. This activity has three parts Request Certification, feasibility Study and Request approval. We have seen various paint application and we have found that it contains old controls which are not updated, we have tried to introduce some new controls to make it more enthralling.

Present System In Use

Present system is all about enhancing the knowledge and skills of a People. Also helps People to learn new Editing skill and also to boost their confidence. It contains image Editor.

People Can Edit there image and can make edit your Image According to your Requirement, and can make Attractive Images Using this Application.

Flaws In Present System

In present system it might be difficult for People to Using Other Image Editor. They may find it difficult to Edit Image and understand so might be they will not be interested in Image Editor.

Need For New System

New system is been prepared keeping in mind that it deals effectively & more flexible than the existing system. New system & technologies may allow faster processing, more storage, and lower costs than ever before. Today, when everything is becoming more and more computerized, one may feel that the task of implementing various things should also be done automatically using the computer programs. Current application is somewhere has its limitations. With a new system we may fill all the required requirements which current application doesn't have. We may be giving videos more frequency and also fast service. With new system we can also improve security and store large amount of data.

The need of new system arises due to the following reasons:

1. To introduce a new objects and functionality in the system.
2. To improve the user experience via fascinating user interface
3. For creating a friendlier environment, than the previous existing system for the users.

Feasibility Study

This is the stage where the feasibility is being studied by the developer for the project. The feasibility study is useful to evaluate the cost and benefit of the new system requested. The feasibility study tries to anticipate future scenarios of the software development. It evaluates the benefits of the new system. The aim of the feasibility study is to see whether it is possible to develop a new system at a reasonable cost. At the end of the feasibility study a decision is taken whether to proceed or not. Over all Feasibility study is the process to determine various solutions of the problem and then picking up one of the best solution. It is measure of how beneficial the development of information system will be to an organization.

The feasibility study has the following objectives:-

- Defines the problem clearly.
- Effective use of resources.
- Analyze the cost and benefit.

A feasibility study looks at the viability of an idea with an emphasis on identifying potential problems. The feasibility study tries to anticipate future scenarios of the software development.

1) Technical feasibility: The technical feasibility always focuses on the existing computer hardware and software. This also includes the need for more hardware and software and the possibility of installing such facility. Technical feasibility concentrates on aspect of project to find out the various hardware and software requirements for the project. In technical feasibility the following issues are taken into consideration. Whether the required technology is available or not. Whether the required resources are available like management programmers, tester & debuggers software and hardware. Once the technical feasibility is established, it is important to consider the monetary factors also, since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this economic feasibility of the proposed system it carried out.

2) Economical Feasibility: Economical analysis is the most frequently used method for evaluating the effectiveness of candidates system. It is also called as cost benefit analysis, the procedure is to determine the benefit and saving that are expected from candidate system and compare them with cost. If benefits outweigh cost then decision is made to design and implement system. It becomes easy to manage such records. Economical Feasibility is helpful for finding the system development cost and benefit, it checks whether it is justifiable or not, for that it checks the software and hardware costs etc.

3) Operational feasibility: Operational feasibility considers the acceptability of the system. It checks whether system will be used if it is developed and implemented and the user of the new system will be able to handle the system or not. The operational feasibility mostly look to the part that how easily the project can be put into the action it looks over that, is there any difficulty on loading of this project or is the project operating smoothly afterwards. This project is operationally feasibility on the screen itself. While making any project it is necessary to study the operational features.

4) Social Feasibility: Social Feasibility is a determination of whether this proposed system will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

- It describes the effect on users from the introduction of the new system considering whether there will be need for retraining the workforce.
- It describes how you propose to ensure user co-operation before changes are introduced.

Project Category

The Project Category is an application called "PhotoShop and Editing Application". This application is designed using python language. From starting page to ending page we have many pages linked together containing different information. Our application is an editing application which helps them for fun and learn simultaneously.

➤ Programming Language Used In Project

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

List of why Python is popular:

- The Python framework also has modules and packages, which facilitates code reusability.
- Python is open source. You can download it for free and use it in your application. You can also read and modify the source code.
- No Compilation of the code – the cycle of Edit-test-debug is fast
- Supports exception handling. Any code is prone to errors. Python generates exceptions that can be handled hence avoids crashing of programs.
- Automatic memory management. Memory management in Python involves a private heap(a data structure that represents a queue) containing all Python objects and data structures.

MySQL

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr,^[12] MediaWiki Twitter and YouTube.

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Online Data Definition Language (DDL) when using the InnoDB Storage Engine.

•Information schema

•Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.[80]

•A set of SQL Mode options to control runtime behaviour, including a strict mode to better adhere to SQL standards.

•X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine

•Transactions with save points when using the default InnoDB Storage Engine.

The NDB Cluster Storage Engine also supports transactions.

•ACID compliance when using InnoDB and NDB Cluster Storage Engines

•SSL support

•Query caching

•Sub-SELECTs (i.e. nested SELECTs)

•Built-in replication support o Asynchronous replication: master-slave from one master to many slaves or many masters to one slave

o Semi synchronous replication: Master to slave replication where the master waits on replication

o Synchronous replication: Multi-master replication is provided in MySQL Cluster

o Virtual Synchronous: Self managed groups of MySQL servers with multi master support can be done using: Galera Cluster or the built in Group Replication plugin[89]

•Full-text indexing and searching[b]

•Embedded database library

- Unicode support[a]
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.[c]
- Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

The developers release minor updates of the MySQL Server approximately every two months. The sources can be obtained from MySQL's website or from MySQL's GitHub repository, both under the GPL license.

Software And Hardware Requirement Specification

Hardware

Hardware, which is abbreviated as HW, refers to all physical components of a computer system, including the devices connected to it. You cannot create a computer or use software without using hardware. The screen on which you are reading this information is also a hardware.

- ✦ RAM :4GB and Above
- ✦ Hard disk : 320 GB and Above
- ✦ Keyboard
- ✦ Mouse

Software

Software is a set of instructions, data, or programs used to operate a computer and execute specific tasks. In simpler terms, software tells a computer how to function.

It's a generic term used to refer to applications, scripts, and programs that run on devices such as PCs, mobile phones, tablets, and other smart devices. Software contrasts with hardware, which is the physical aspects of a computer that perform the work.

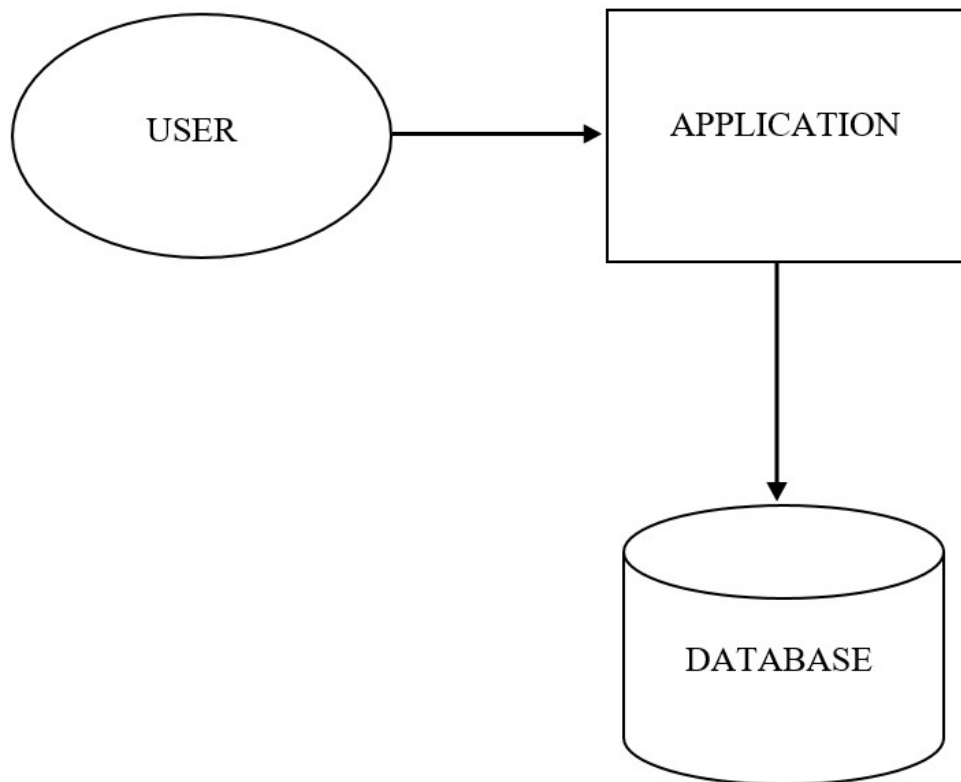
- Microsoft Window 11

Operating System

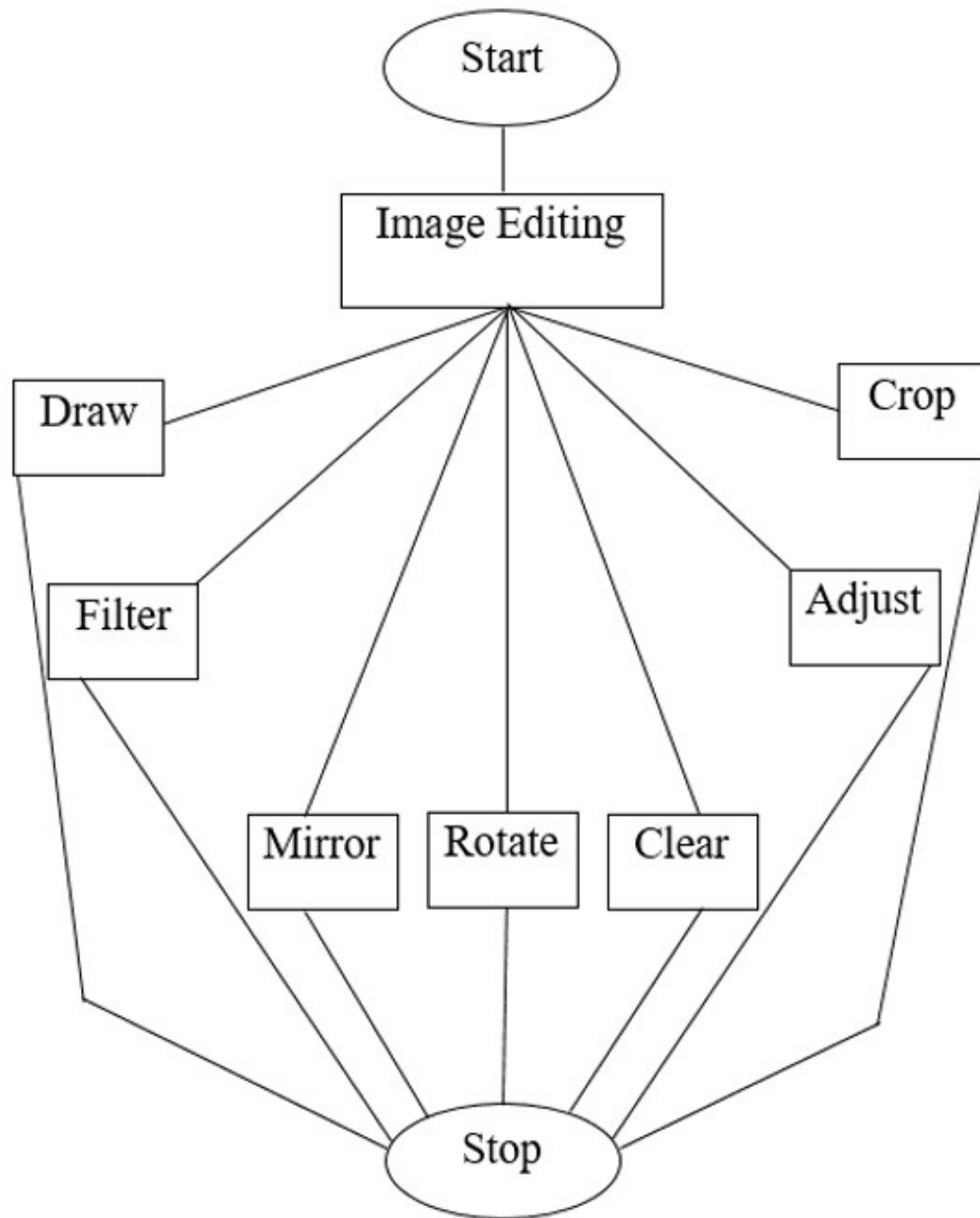
- Windows 11

Detailed System Analysis

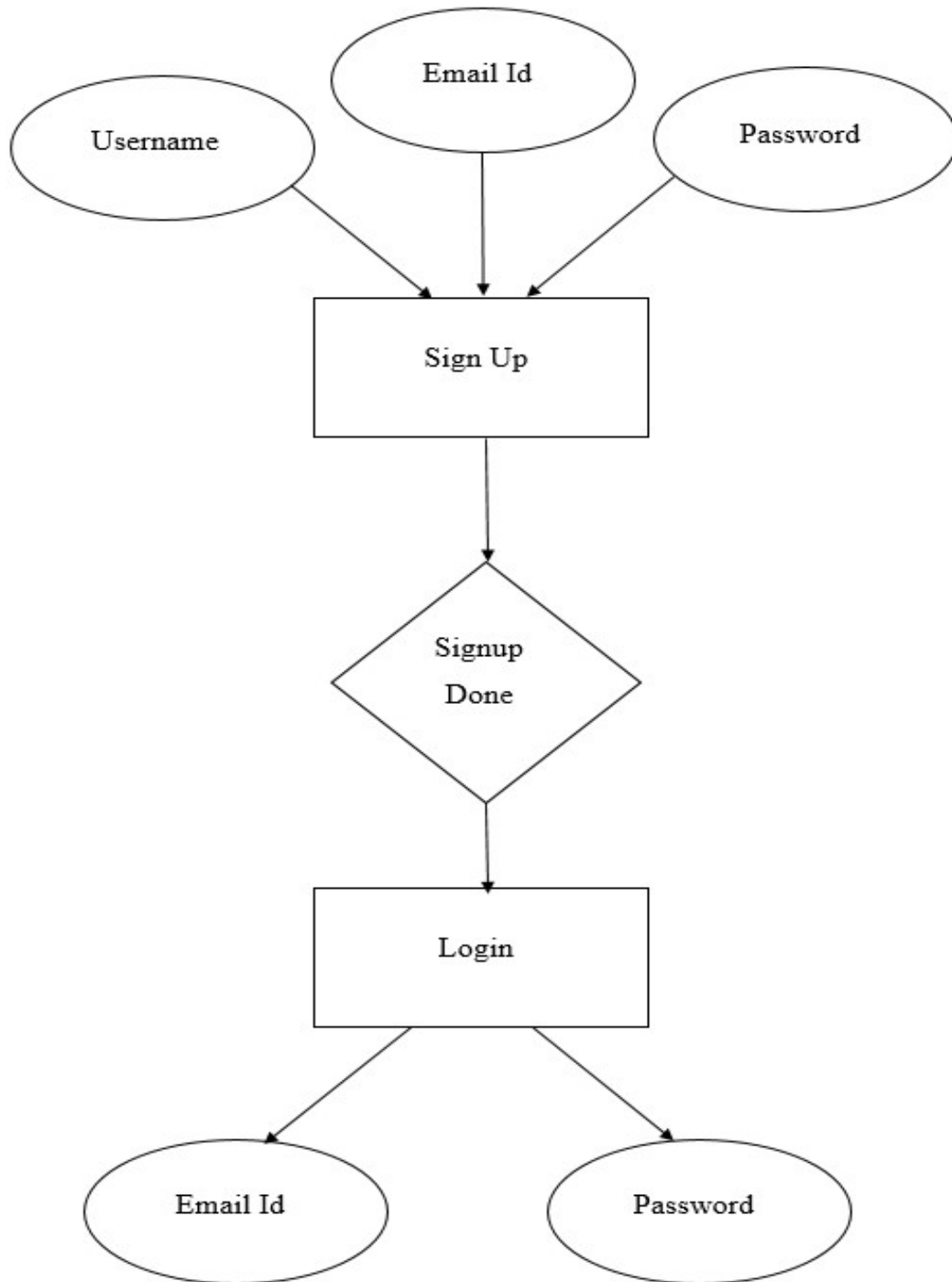
Data Flow Diagram



Data System Analysis



Entity Relationship Diagram



DATA STRUCTURE AND TABLES

```
MySQL 8.0 Command Line Client
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use userdata;
Database changed
mysql> select * from data;
```

id	email	Username	Password
1	krutika	pagal	123456
2	hffhdulhf	ssufhskfdsk	2222
3	g	t	222
4	krutikabansod	bansodkrutika3@gmail.com	123456
5	krutika@301	Googly	2002
6	krutika1	googly1	2001
7	k	g	2001
8	sidhi	krutika	googly
9	ganesh	tathe	2001
10	krutika	bansod	2001
11	krutikabansod1@gmail.com	akanksha	googly
12	krutikabansod2@gmail.com	kruti	2001
13	krutikabansod3@gmail.com	moogly	2001

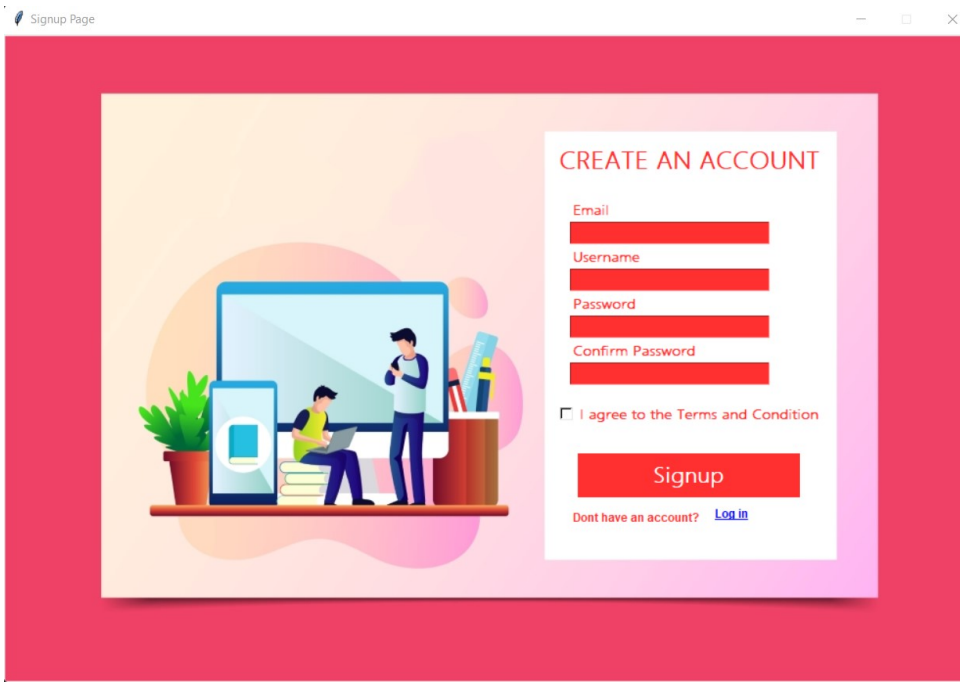
```
13 rows in set (0.05 sec)

mysql>
```

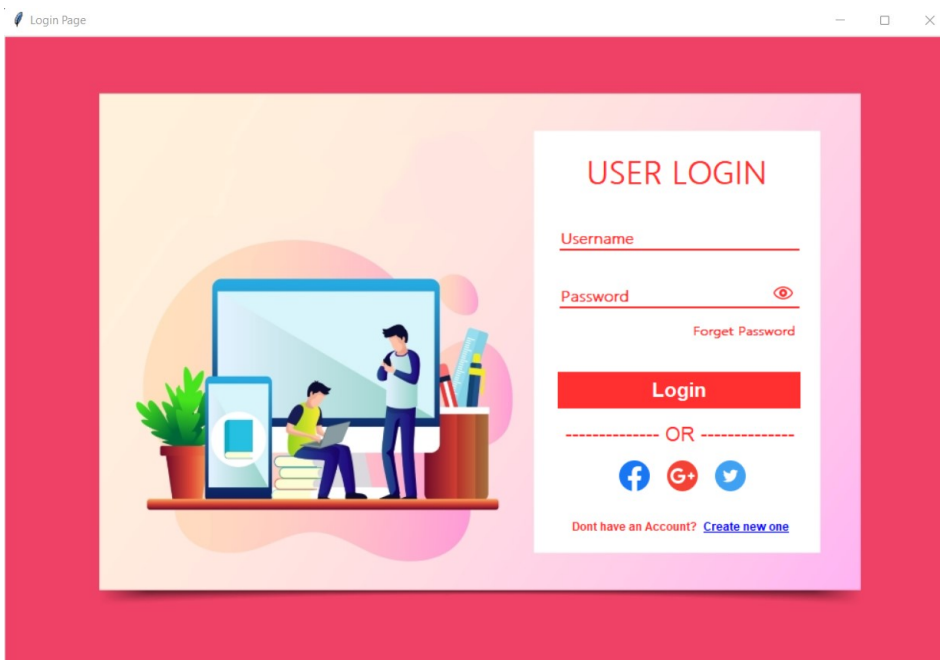
System Design

FORM DESIGN

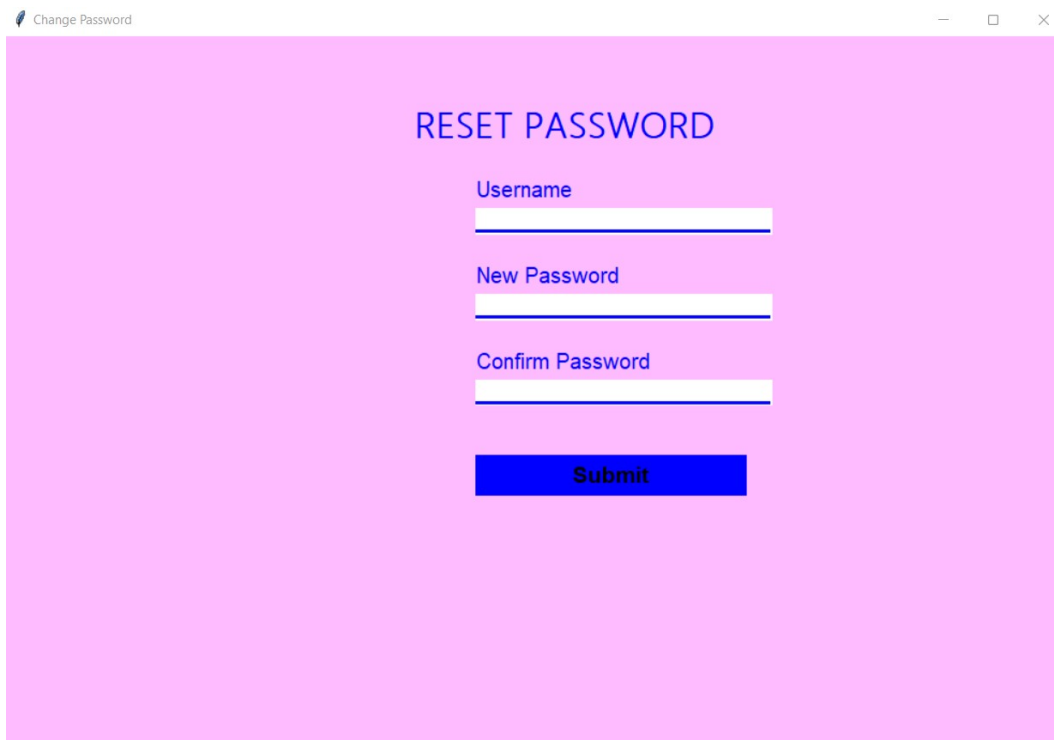
REGISTRATION FORM



LOGIN FORM



CHANGE PASSWORD



Change Password

RESET PASSWORD

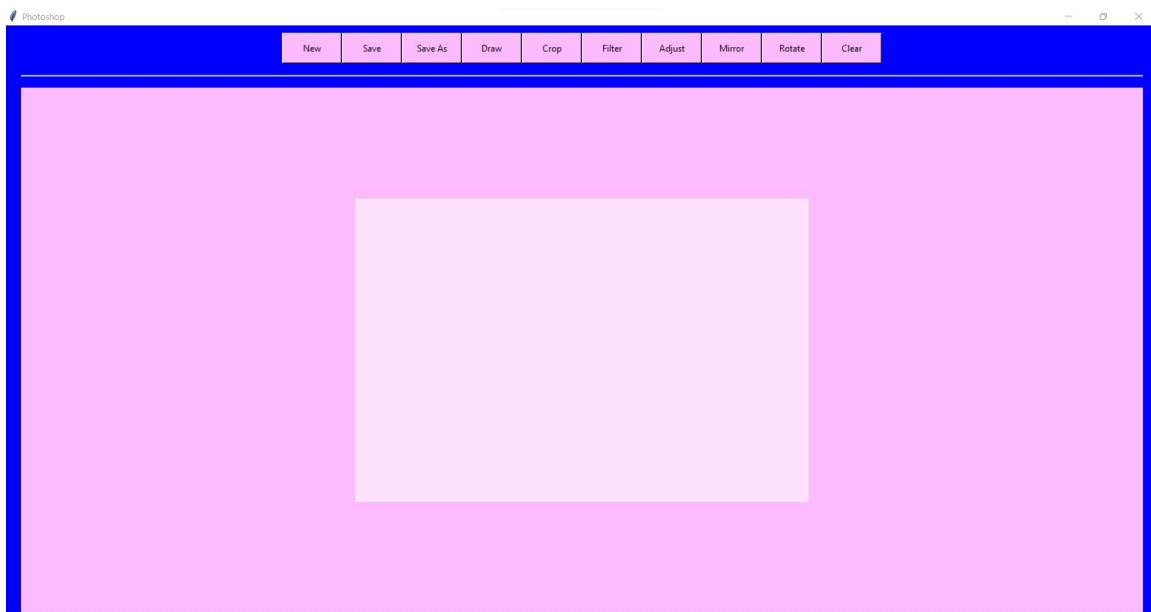
Username

New Password

Confirm Password

Submit

IMAGE EDITING



SOURCE CODE

REGISTRATION CODE

```
from tkinter import*
from tkinter import messagebox
from PIL import ImageTk
import pymysql

def clear():
    emailEntry.delete(0,END)
    usernameEntry.delete(0,END)
    passwordEntry.delete(0,END)
    confirmEntry.delete(0,END)
    check.set(0)

def connect_Database():
    if emailEntry.get()==" or usernameEntry.get()==" or
passwordEntry.get()==" or confirmEntry.get()=="":
        messagebox.showerror('Error','All Fields Are Required')
    elif passwordEntry.get()!=confirmEntry.get():
        messagebox.showerror('Error','Password Mismatch')
    elif check.get()==0:
        messagebox.showerror('Error','Please accept Terms & Conditions')
    else:
        try:

con=pymysql.connect(host='localhost',user='root',password='03122001')
        mycursor=con.cursor()
        except:
            messagebox.showerror('Error','Database Connectivity Issue, Please
Try Again')
            return
        try:
            query='create database userdata'
            mycursor.execute(query)
            query='use userdata'
            mycursor.execute(query)
```

```
query='create table data(id int auto_increment primary key not null, email
varchar(50),Username varchar(90),Password varchar(50))'
```

```
mycursor.execute(query)
```

```
except:
```

```
    mycursor.execute('use userdata')
```

```
query='select * from data where Username=%s'
```

```
mycursor.execute(query,(usernameEntry.get()))
```

```
row=mycursor.fetchone()
```

```
if row !=None:
```

```
    messagebox.showerror('Error','Username Already exists')
```

```
else:
```

```
    query='insert into data(email,username,password) values(%s,%s,%s)'
```

```
mycursor.execute(query,(emailEntry.get(),usernameEntry.get(),passwordEnt
ry.get()))
```

```
    con.commit()
```

```
    con.close()
```

```
    messagebox.showinfo('Success','Registration is successful')
```

```
    clear()
```

```
    signup_window.destroy()
```

```
    import Signup
```

```
def Login_page():
```

```
    signup_window.destroy()
```

```
    import Signup
```

```
signup_window=Tk()
```

```
signup_window.title('Signup Page')
```

```
signup_window.resizable(False,False)
```

```
background=ImageTk.PhotoImage(file='bg.jpg')
```

```
bgLabel=Label(signup_window,image=background)
```

```
bgLabel.grid()
```

```
frame=Frame(signup_window,bg='white')  
frame.place(x=554,y=100)
```

```
heading=Label(frame,text='CREATE AN ACCOUNT',font=('Microsoft  
Yahei Ui Light',18,'bold'),bg='white',fg='firebrick1')  
heading.grid(row=0,column=0,padx=10,pady=10)
```

```
emailLabel=Label(frame,text='Email',font=('Microsoft Yahei Ui  
Light',10,'bold'),bg='white',fg='firebrick1')  
emailLabel.grid(row=1,column=0,sticky='w',padx=25,pady=(10,0))
```

```
emailEntry=Entry(frame,width=25,font=('Microsoft Yahei Ui  
Light',10,'bold'),fg='white',bg='firebrick1')  
emailEntry.grid(row=2,column=0,sticky='w',padx=25)
```

```
usernameLabel=Label(frame,text='Username',font=('Microsoft Yahei Ui  
Light',10,'bold'),bg='white',fg='firebrick1')  
usernameLabel.grid(row=3,column=0,sticky='w',padx=25)
```

```
usernameEntry=Entry(frame,width=25,font=('Microsoft Yahei Ui  
Light',10,'bold'),fg='white',bg='firebrick1')  
usernameEntry.grid(row=4,column=0,sticky='w',padx=25)
```

```
passwordLabel=Label(frame,text='Password',font=('Microsoft Yahei Ui  
Light',10,'bold'),bg='white',fg='firebrick1')  
passwordLabel.grid(row=5,column=0,sticky='w',padx=25)
```

```
passwordEntry=Entry(frame,width=25,font=('Microsoft Yahei Ui  
Light',10,'bold'),fg='white',bg='firebrick1')  
passwordEntry.grid(row=6,column=0,sticky='w',padx=25)
```

```
confirmLabel=Label(frame,text='Confirm Password',font=('Microsoft Yahei  
Ui Light',10,'bold'),bg='white',  
fg='firebrick1')  
confirmLabel.grid(row=7,column=0,sticky='w',padx=25)
```

```
confirmEntry=Entry(frame,width=25,font=('Microsoft Yahei Ui  
Light',10,'bold'),fg='white',bg='firebrick1')  
confirmEntry.grid(row=8,column=0,sticky='w',padx=25)
```

```
check=IntVar()
```

```
termsandcondition=Checkbutton(frame,text='I agree to the Terms and  
Condition',font=('Microsoft Yahei Ui Light',10,  
'bold'),bg='white',fg='firebrick1',activebackground='white',activeforeground  
='firebrick1',cursor='hand2',variable=check)  
termsandcondition.grid(row=9,column=0,padx=10,pady=15)
```

```
signupButton=Button(frame,text='Signup',font=('Microsoft Yahei Ui  
Light',16,'bold'),bd=0,bg='firebrick1',  
fg='white',activebackground='firebrick1',activeforeground='white',width=17,  
command=connect_Database)  
signupButton.grid(row=10,column=0,pady=10)
```

```
alreadyaccount=Label(frame,text="Dont have an account?",font=('Open  
Sans','9','bold'),bg='white',fg='firebrick1')  
alreadyaccount.grid(row=11,column=0,sticky='w',padx=25)
```

```
alreadyaccount=Label(frame,text="Dont have an account?",font=('Open  
Sans','9','bold'),bg='white',fg='firebrick1')  
alreadyaccount.grid(row=11,column=0,sticky='w',padx=25)
```

```
LoginButton=Button(frame,text='Log in',font=('Open Sans','9','bold  
underline'),  
bg='white',fg='blue', bd=0,  
cursor='hand2',activebackground='white',activeforeground='blue',command=  
Login_page)  
LoginButton.place(x=170,y=380)
```

```
signup_window.mainloop()
```


LOGIN CODE

```
from tkinter import*
from tkinter import messagebox
from PIL import ImageTk
import pymysql

#Functionality Part

def forget_pass():
    def change_password():
        if user_entry.get()==" or newpass_entry.get()==" or
        confirmpass_entry.get()=="":
            messagebox.showerror('Error','All fields Are
            Required',parent=window,bgImage="log.jpg")

        elif newpass_entry.get()!=confirmpass_entry.get():
            messagebox.showerror('Error','Password and Confirm Password are
            not Matching',parent=window)
        else:
            con=pymysql.connect(host='localhost',user ='root',password
            ='03122001',database ='userdata')
            mycursor=con.cursor()
            query = 'select * from data where username=%s'
            mycursor.execute(query,(user_entry.get()))
            row=mycursor.fetchone()

            if row==None:
                messagebox.showerror('Error','Incorrect username',parent=window)
            else:
                query='update data set password=%s where username=%s'
                mycursor.execute(query,(newpass_entry.get(),user_entry.get()))
                con.commit()
                con.close()
                messagebox.showerror('Error','Password is reset,Please Login with
                new password',parent=window)
                window.destroy()

    window = Toplevel()
```

```
window.title('Change Password')
```

```
Login_window.geometry('990x660+50+50')
```

```
bgPic = ImageTk.PhotoImage(file='bg.jpg')  
bgLabel = Label(window,image=bgPic,bg='plum1')  
bgLabel.grid()
```

```
heading_Label=Label(window,text='RESET PASSWORD',bg=  
'plum1',font=('Microsoft Yahei Ui Light',23,'bold'),  
fg='blue')  
heading_Label.place(x=380,y=60)
```

```
userLabel=Label(window,text='Username',font=14,bd=0,bg='plum1',fg='blue')  
userLabel.place(x=440,y=130)
```

```
user_entry=Entry(window,width=25,font=14,bd=0,fg='blue')  
user_entry.place(x=440,y=160)
```

```
Frame(window, width=275, height=3, bg='blue').place(x=440,y=180)
```

```
passwordLabel=Label(window,text='New  
Password',font=14,bd=0,bg='plum1',fg='blue')  
passwordLabel.place(x=440,y=210)
```

```
newpass_entry=Entry(window,width=25,font=14,bd=0,fg='blue')  
newpass_entry.place(x=440,y=240)
```

```
Frame(window, width=275, height=3, bg='blue').place(x=440,y=260)
```

```
confirmpass_Label=Label(window,text='Confirm  
Password',font=14,bg='plum1',bd=0,fg='blue')  
confirmpass_Label.place(x=440,y=290)
```

```

confirmpass_entry=Entry(window,width=25,font=('arial',14,'bold'),bd=0,fg=
'blue')
confirmpass_entry.place(x=440,y=320)

Frame(window, width=275, height=3, bg='blue').place(x=440,y=340)

SubmitButton=Button(window,width=19,text='Submit',font=('Open
Sans','16','bold'),
bd=0,bg="blue",
cursor='hand2',activebackground='blue',activeforeground='white',command=
change_password)
SubmitButton.place(x=440,y=390)

def Login_user():
    if usernameEntry.get()==" or passwordEntry.get()=="":
        messagebox.showerror('Error','All Fields Are Required')

    else:
        try:

con=pymysql.connect(host='localhost',user='root',password='03122001')
    mycursor=con.cursor()
    except:
        messagebox.showerror('Error','Connection is not established try
again')
        return
    query='use userdata'
    mycursor.execute(query)
    query='select * from data where username=%s and password=%s'
    mycursor.execute(query,(usernameEntry.get(),passwordEntry.get()))
    row=mycursor.fetchone()

    if row==None:
        messagebox.showerror('Error','Invalid Username or password')
    else:
        messagebox.showinfo('Welcome','Login is Successful')

```

```

def signup_page():
    Login_window.destroy()
    import register

def hide():
    openeye.config(file='closeeye.png')
    passwordEntry.config(show='*')
    eyeButton.config(command=show)

def show():
    openeye.config(file='openeye.png')
    passwordEntry.config(show='')
    eyeButton.config(command=hide)

def user_enter(event):
    if usernameEntry.get()=='Username':
        usernameEntry.delete(0,END)

def password_enter(event):
    if passwordEntry.get()=='Password':
        passwordEntry.delete(0,END)

#GUI Part
Login_window=Tk()
Login_window.geometry('990x660+50+50')
#Login_window.resizable(0,0)
Login_window.title('Login Page')
bgImage=ImageTk.PhotoImage(file='bg.jpg')

bgLabel=Label(Login_window,image=bgImage)
bgLabel.place(x=0,y=0)

heading=Label(Login_window,text='USER LOGIN',font=('Microsoft Yahei
Ui Light',23,'bold'),bg='white',fg='firebrick1')
heading.place(x=605,y=120)

usernameEntry=Entry(Login_window,width=25,font=('Microsoft Yahei Ui
Light',11,'bold'),bd=0,fg='firebrick1')
usernameEntry.place(x=580,y=200)

```

```

usernameEntry.insert(0,'Username')

usernameEntry.bind('<FocusIn>',user_enter)

frame1=Frame(Login_window,width=250,height=2,bg='firebrick1')
frame1.place(x=580,y=222)

passwordEntry=Entry(Login_window,width=25,font=('Microsoft Yahei Ui
Light',11,'bold'),bd=0,fg='firebrick1')
passwordEntry.place(x=580,y=260)
passwordEntry.insert(0,'Password')

passwordEntry.bind('<FocusIn>',password_enter)

frame2=Frame(Login_window,width=250,height=2,bg='firebrick1')
frame2.place(x=580,y=282)

openeye=PhotoImage(file='openeye.png')
eyeButton=Button(Login_window,image=openeye,bd=0,bg='white',activeba
ckground='white',cursor='hand2',command=hide)
eyeButton.place(x=800,y=255)

forgetButton=Button(Login_window,text='Forget
Password',bd=0,bg='white',activebackground='white',cursor='hand2',
font=('Microsoft Yahei Ui
Light',9,'bold'),fg='firebrick1',activeforeground='firebrick1',command=forge
t_pass)
forgetButton.place(x=715,y=295)

loginButton=Button(Login_window,text='Login',font=('Open
Sans',16,'bold'),fg='white',bg='firebrick1',
activebackground='white',cursor='hand2',bd=0,width=19,command=Login_
user)
loginButton.place(x=578,y=350)

orLabel=Label(Login_window,text='----- OR -----
',font=('Open Sans',16),fg='firebrick1',bg='white')
orLabel.place(x=583,y=400)

```

```
facebook_logo=PhotoImage(file='facebook.png')
fbLabel=Label(Login_window,image=facebook_logo,bg='white')
fbLabel.place(x=640,y=440)
```

```
google_logo=PhotoImage(file='google.png')
googleLabel=Label(Login_window,image=google_logo,bg='white')
googleLabel.place(x=690,y=440)
```

```
twitter_logo=PhotoImage(file='twitter.png')
twitterLabel=Label(Login_window,image=twitter_logo,bg='white')
twitterLabel.place(x=740,y=440)
```

```
signupLabel=Label(Login_window,text='Dont have an
Account?',font=('Open Sans',9,'bold'),fg='firebrick1',bg='white')
signupLabel.place(x=590,y=500)
```

```
newaccountButton=Button(Login_window,text='Create new
one',font=('Open Sans',9,'bold underline'),
bg='white',fg='blue',bd=0,
cursor='hand2',activebackground='white',activeforeground='blue',command=
signup_page)
newaccountButton.place(x=727,y=500)
```

```
Login_window.mainloop()
```

IMAGE EDITING CODE

MAIN.PY

```
import tkinter as tk
from tkinter import ttk
from editBar import EditBar
from imageViewer import ImageViewer

class Main(tk.Tk):

    def __init__(self):
        tk.Tk.__init__(self)

        self.filename = ""
        self.original_image = None
        self.processed_image = None
        self.is_image_selected = False
        self.is_draw_state = False
        self.is_crop_state = False
        self.is_mirror_state = False
        self.is_rotate_state = False
        self.filter_frame = None
        self.adjust_frame = None
        self.configure(bg="blue")
        self.title("Photoshop")

        self.editbar = EditBar(master=self)
        separator1 = ttk.Separator(master=self, orient=tk.HORIZONTAL)
        self.image_viewer = ImageViewer(master=self,)

        self.editbar.pack(pady=10)
        separator1.pack(fill=tk.X, padx=20, pady=5)
        self.image_viewer.pack(fill=tk.BOTH, padx=20, pady=10, expand=1)
```

IMAGEVIEWER.PY

```
from tkinter import Frame, Canvas, CENTER, ROUND
from PIL import Image, ImageTk
import cv2
```

```
class ImageViewer(Frame):
```

```
    def __init__(self, master=None):
        Frame.__init__(self, master=master, bg="plum1", width=600,
            height=400)

        self.shown_image = None
        self.x = 0
        self.y = 0
        self.crop_start_x = 0
        self.crop_start_y = 0
        self.crop_end_x = 0
        self.crop_end_y = 0
        self.draw_ids = list()
        self.rectangle_id = 0
        self.ratio = 0

        self.canvas = Canvas(self, bg="thistle1", width=600, height=400)
        self.canvas.place(relx=0.5, rely=0.5, anchor=CENTER)

    def show_image(self, img=None):
        self.clear_canvas()

        if img is None:
            image = self.master.processed_image.copy()
        else:
            image = img

        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        height, width, channels = image.shape
        ratio = height / width
```



```

new_width = width
new_height = height

if height > self.winfo_height() or width > self.winfo_width():
    if ratio < 1:
        new_width = self.winfo_width()
        new_height = int(new_width * ratio)
    else:
        new_height = self.winfo_height()
        new_width = int(new_height * (width / height))

self.shown_image = cv2.resize(image, (new_width, new_height))
self.shown_image =
ImageTk.PhotoImage(Image.fromarray(self.shown_image))

self.ratio = height / new_height

self.canvas.config(width=new_width, height=new_height)
self.canvas.create_image(new_width / 2, new_height / 2,
anchor=CENTER, image=self.shown_image)

def activate_draw(self):
    self.canvas.bind("<ButtonPress>", self.start_draw)
    self.canvas.bind("<B1-Motion>", self.draw)

    self.master.is_draw_state = True

def activate_crop(self):
    self.canvas.bind("<ButtonPress>", self.start_crop)
    self.canvas.bind("<B1-Motion>", self.crop)
    self.canvas.bind("<ButtonRelease>", self.end_crop)

    self.master.is_crop_state = True

def activate_mirror(self):
    self.canvas.bind("<ButtonPress>", self.start_mirror())
    self.canvas.bind("<B1-Motion>", self.mirror)

```

```

self.canvas.bind("<ButtonRelease>", self.end_mirror)

    self.master.is_crop_state = True

def deactivate_draw(self):
    self.canvas.unbind("<ButtonPress>")
    self.canvas.unbind("<B1-Motion>")

    self.master.is_draw_state = False

def deactivate_crop(self):
    self.canvas.unbind("<ButtonPress>")
    self.canvas.unbind("<B1-Motion>")
    self.canvas.unbind("<ButtonRelease>")

    self.master.is_crop_state = False

def start_draw(self, event):
    self.x = event.x
    self.y = event.y

def draw(self, event):
    self.draw_ids.append(self.canvas.create_line(self.x, self.y, event.x,
event.y, width=2,
                                                fill="red", capstyle=ROUND,
smooth=True))

    cv2.line(self.master.processed_image, (int(self.x * self.ratio), int(self.y
* self.ratio)),
            (int(event.x * self.ratio), int(event.y * self.ratio)),
            (0, 0, 255), thickness=int(self.ratio * 2),
            lineType=8)

    self.x = event.x
    self.y = event.y

def start_rotate(self, event):
    self.x = event.x

```

```
self.y = event.y
```

```
def start_mirror(self, event):  
    self.x = event.x  
    self.y = event.y
```

```
def start_crop(self, event):  
    self.crop_start_x = event.x  
    self.crop_start_y = event.y
```

```
def crop(self, event):  
    if self.rectangle_id:  
        self.canvas.delete(self.rectangle_id)
```

```
    self.crop_end_x = event.x  
    self.crop_end_y = event.y
```

```
    self.rectangle_id = self.canvas.create_rectangle(self.crop_start_x,  
self.crop_start_y,  
                                                    self.crop_end_x, self.crop_end_y,  
width=1)
```

```
def end_crop(self, event):  
    if self.crop_start_x <= self.crop_end_x and self.crop_start_y <=  
self.crop_end_y:  
        start_x = int(self.crop_start_x * self.ratio)  
        start_y = int(self.crop_start_y * self.ratio)  
        end_x = int(self.crop_end_x * self.ratio)  
        end_y = int(self.crop_end_y * self.ratio)  
    elif self.crop_start_x > self.crop_end_x and self.crop_start_y <=  
self.crop_end_y:  
        start_x = int(self.crop_end_x * self.ratio)  
        start_y = int(self.crop_start_y * self.ratio)  
        end_x = int(self.crop_start_x * self.ratio)  
        end_y = int(self.crop_end_y * self.ratio)  
    elif self.crop_start_x <= self.crop_end_x and self.crop_start_y >  
self.crop_end_y:
```

```
start_x = int(self.crop_start_x * self.ratio)
    start_y = int(self.crop_end_y * self.ratio)
    end_x = int(self.crop_end_x * self.ratio)
    end_y = int(self.crop_start_y * self.ratio)
else:
    start_x = int(self.crop_end_x * self.ratio)
    start_y = int(self.crop_end_y * self.ratio)
    end_x = int(self.crop_start_x * self.ratio)
    end_y = int(self.crop_start_y * self.ratio)

x = slice(start_x, end_x, 1)
y = slice(start_y, end_y, 1)

self.master.processed_image = self.master.processed_image[y, x]

self.show_image()

def clear_canvas(self):
    self.canvas.delete("all")

def clear_draw(self):
    self.canvas.delete(self.draw_ids)
```

ADJUSTFRAME.PY

```
from tkinter import Toplevel, Label, Scale, Button, HORIZONTAL, RIGHT
import cv2
import numpy as np
```

```
s = 100
```

```
MAX_VALUE = 100
```

```
class AdjustFrame(Toplevel):
```

```
    def __init__(self, master=None):
```

```
        Toplevel.__init__(self, master=master, bg="red")
```

```
        self.brightness_value = 255
```

```
        self.previous_brightness_value = 255
```

```
        self.i=0
```

```
        self.contrast_value = 127
```

```
        self.previous_contrast_value = 127
```

```
        self.original_image = self.master.processed_image
```

```
        self.copy=self.original_image
```

```
        self.copy1=cv2.cvtColor(self.original_image, cv2.COLOR_BGR2HLS)
```

```
        self.processing_image = self.master.processed_image
```

```
        self.contrast_label = Label(self, text="Contrast")
```

```
        self.contrast_scale = Scale(self, from_=0, to_=255, length=250,
```

```
resolution=1,
```

```
orient=HORIZONTAL, background="powder blue")
```

```
        self.brightness_label = Label(self, text="Brightness")
```

```
        self.brightness_scale = Scale(self, from_=0, to_=510, length=250,
```

```
resolution=1,
```

```
orient=HORIZONTAL)
```

```
        # self.saturation_label = Label(self, text="Saturation")
```

```
        # self.saturation_scale = Scale(self, from_=0, to_=2, length=250,
```

```
resolution=0.1,
```

```
        # orient=HORIZONTAL)
```

```
        self.clarity_label = Label(self, text="Blur")
```

```

self.clarity_scale = Scale(self, from_=0, to_=2, length=250,
resolution=0.1,orient=HORIZONTAL)
    self.warmth_label = Label(self, text="Warmth")
    self.warmth_scale = Scale(self, from_=0, to_=1, length=250,
resolution=0.05,orient=HORIZONTAL)
    self.cool_label = Label(self, text="Cool")
    self.cool_scale = Scale(self, from_=0, to_=1, length=250,
resolution=0.05,orient=HORIZONTAL)

self.r_label = Label(self, text="R")
self.r_scale = Scale(self, from_=-100, to_=100, length=250,
resolution=1,
orient=HORIZONTAL)
self.g_label = Label(self, text="G")
self.g_scale = Scale(self, from_=-100, to_=100, length=250,
resolution=1,
orient=HORIZONTAL)
self.b_label = Label(self, text="B")
self.b_scale = Scale(self, from_=-100, to_=100, length=250,
resolution=1,
orient=HORIZONTAL)

self.apply_button = Button(self, text="Apply",width=15,bg="powder
blue")
self.preview_button = Button(self,
text="Preview",width=15,bg="powder blue")
self.cancel_button = Button(self, text="Cancel",width=15,bg="powder
blue")

self.brightness_scale.set(255)
self.contrast_scale.set(127)
self.warmth_scale.set(0)
self.cool_scale.set(0)
# self.saturation_scale.set(1)
self.clarity_scale.set(0)

self.apply_button.bind("<ButtonRelease>", self.apply_button_released)

```

```
self.preview_button.bind("<ButtonRelease>", self.show_button_release)
```

```
self.cancel_button.bind("<ButtonRelease>", self.cancel_button_released)
```

```
self.brightness_label.pack()
self.brightness_scale.pack()
self.warmth_label.pack()
self.warmth_scale.pack()
self.cool_label.pack()
self.cool_scale.pack()
self.clarity_label.pack()
self.clarity_scale.pack()
self.contrast_label.pack()
self.contrast_scale.pack()
self.r_label.pack()
self.r_scale.pack()
self.g_label.pack()
self.g_scale.pack()
self.b_label.pack()
self.b_scale.pack()
self.cancel_button.pack()
self.preview_button.pack()
self.apply_button.pack()
```

```
def apply_button_released(self, event):
    self.show_button_release(self)
    self.master.processed_image = self.processing_image
    self.close()
```

```
def gamma_function(self, channel, gamma):
    invGamma = 1 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
                      for i in np.arange(0, 256)]).astype("uint8") # creating
lookup table
    channel = cv2.LUT(channel, table)
    return channel
```

```
def show_button_release(self, event):
```

```

temp = self.copy.copy
    self.show_image(self.copy)
    self.original_image = self.copy

b, g, r = cv2.split(self.processing_image)

    for b_value in b:
        cv2.add(b_value, self.b_scale.get(), b_value)
    for g_value in g:
        cv2.add(g_value, self.g_scale.get(), g_value)
    for r_value in r:
        cv2.add(r_value, self.r_scale.get(), r_value)

self.rgb = cv2.merge((b, g, r))

    brightness = int((self.brightness_scale.get() - 0) * (255 - (-255)) / (510 -
0) + (-255))

    contrast = int((self.contrast_scale.get() - 0) * (127 - (-127)) / (254 - 0) +
(-127))

    if brightness != 0:

        if brightness > 0:

            shadow = brightness

            max = 255

        else:

            shadow = 0
            max = 255 + brightness

        al_pha = (max - shadow) / 255

```



```

ga_mma = shadow

    # The function addWeighted calculates
    # the weighted sum of two arrays
    cal = cv2.addWeighted(self.rgb, al_pha,

self.rgb, 0, ga_mma)

else:
    cal = self.rgb

if contrast != 0:
    Alpha = float(131 * (contrast + 127)) / (127 * (131 - contrast))
    Gamma = 127 * (1 - Alpha)

    # The function addWeighted calculates
    # the weighted sum of two arrays
    cal = cv2.addWeighted(cal, Alpha,
                          cal, 0, Gamma)

clar = self.clarity_scale.get()
print(clar)
if clar!=0:
    clar = (int)(clar * 10)
    img = cv2.blur(cal, (clar, clar))
    self.processing_image = img
    # cv2.imshow("test", img)
    # cv2.waitKey(0)
else:
    self.processing_image = cal

warmth=self.warmth_scale.get()
warmth/=2
img = self.processing_image
img[:, :, 0] = self.gamma_function(img[:, :, 0], 1-warmth) # down
scaling blue channel

```

```

img[:, :, 2] = self.gamma_function(img[:, :, 2], 1+warmth) # up scaling red
channel
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    hsv[:, :, 1] = self.gamma_function(hsv[:, :, 1], 1+warmth-0.01) # up
scaling saturation channel
    img = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    self.processing_image = img

cool = self.cool_scale.get()
cool /= 2
img1 = self.processing_image
img1[:, :, 0] = self.gamma_function(img1[:, :, 0], 1 + cool) # down
scaling blue channel
img1[:, :, 2] = self.gamma_function(img1[:, :, 2], 1 - cool) # up scaling
red channel
hsv1 = cv2.cvtColor(img1, cv2.COLOR_BGR2HSV)
hsv1[:, :, 1] = self.gamma_function(hsv1[:, :, 1], 1 - cool+0.01) # up
scaling saturation channel
img1 = cv2.cvtColor(hsv1, cv2.COLOR_HSV2BGR)
self.processing_image = img1

self.original_image=temp
self.processing_image=cv2.addWeighted(self.processing_image,0.3,self.rgb,
0.7,0.0)
# self.processing_image = cv2.addWeighted(self.processing_image,
0.3, img, 0.7, 0.0)
self.show_image(self.processing_image)
# self.destroy()

def cancel_button_released(self, event):
    self.close()

def show_image(self, img=None):
    self.master.image_viewer.show_image(img=img)

def close(self):
    self.show_image()
    self.destroy()

```

EDITBAR.PY

```
import tkinter as tk
from tkinter import Frame, Button, LEFT
from tkinter import filedialog
from filterFrame import FilterFrame
from adjustFrame import AdjustFrame
import cv2
```

```
class EditBar(Frame):
```

```
    def __init__(self, master=None):
```

```
        Frame.__init__(self, master=master)
```

```
        self.new_button = Button(self,
text="New",bg="plum1",width="10",height="2")
        self.save_button = Button(self,
text="Save",bg="plum1",width="10",height="2")
        self.save_as_button = Button(self, text="Save
As",bg="plum1",width="10",height="2")
        self.draw_button = Button(self,
text="Draw",bg="plum1",width="10",height="2")
        self.crop_button = Button(self,
text="Crop",bg="plum1",width="10",height="2")
        self.mirror_button = Button(self,
text="Mirror",bg="plum1",width="10",height="2")
        self.rotate_button = Button(self,
text="Rotate",bg="plum1",width="10",height="2")
        self.filter_button = Button(self,
text="Filter",bg="plum1",width="10",height="2")
        self.adjust_button = Button(self,
text="Adjust",bg="plum1",width="10",height="2")
        self.clear_button = Button(self,
text="Clear",bg="plum1",width="10",height="2")
```

```
        self.new_button.bind("<ButtonRelease>", self.new_button_released)
```

```
        self.save_button.bind("<ButtonRelease>", self.save_button_released)
```

```

self.save_as_button.bind("<ButtonRelease>", self.save_as_button_released)
self.draw_button.bind("<ButtonRelease>", self.draw_button_released)
self.crop_button.bind("<ButtonRelease>", self.crop_button_released)
self.mirror_button.bind("<ButtonRelease>",
self.mirror_button_released)
self.rotate_button.bind("<ButtonRelease>", self.rotate_button_released)
self.filter_button.bind("<ButtonRelease>", self.filter_button_released)
self.adjust_button.bind("<ButtonRelease>",
self.adjust_button_released)
self.clear_button.bind("<ButtonRelease>", self.clear_button_released)

```

```

self.new_button.pack(side=LEFT)
self.save_button.pack(side=LEFT)
self.save_as_button.pack(side=LEFT)
self.draw_button.pack(side=LEFT)
self.crop_button.pack(side=LEFT)
self.filter_button.pack(side=LEFT)
self.adjust_button.pack(side=LEFT)
self.mirror_button.pack(side=LEFT)
self.rotate_button.pack(side=LEFT)
self.clear_button.pack()

```

```

def rotate_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.rotate_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()

        self.master.processed_image =
cv2.rotate(self.master.original_image, cv2.ROTATE_90_CLOCKWISE)
        self.master.image_viewer.show_image()
        self.master.original_image = self.master.processed_image

def mirror_button_released(self, event):

```

```

if self.winfo_containing(event.x_root, event.y_root) == self.mirror_button:
    if self.master.is_image_selected:
        if self.master.is_draw_state:
            self.master.image_viewer.deactivate_draw()
        if self.master.is_crop_state:
            self.master.image_viewer.deactivate_crop()

        self.master.processed_image =
cv2.flip(self.master.processed_image, 1)
        self.master.image_viewer.show_image()
        self.master.original_image = self.master.processed_image

```

```

def new_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.new_button:
        if self.master.is_draw_state:
            self.master.image_viewer.deactivate_draw()
        if self.master.is_crop_state:
            self.master.image_viewer.deactivate_crop()

```

```

filename = filedialog.askopenfilename()
image = cv2.imread(filename)

```

```

if image is not None:
    self.master.filename = filename
    self.master.original_image = image.copy()
    self.master.processed_image = image.copy()
    self.master.image_viewer.show_image()
    self.master.is_image_selected = True

```

```

def save_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.save_button:
        if self.master.is_image_selected:
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()
            if self.master.is_draw_state:

```

```

self.master.image_viewer.deactivate_draw()

save_image = self.master.processed_image
image_filename = self.master.filename
cv2.imwrite(image_filename, save_image)

def save_as_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.save_as_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()

            original_file_type = self.master.filename.split('.')[-1]
            filename = filedialog.asksaveasfilename()
            filename = filename + "." + original_file_type

            save_image = self.master.processed_image
            cv2.imwrite(filename, save_image)

            self.master.filename = filename

def draw_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.draw_button:
        if self.master.is_image_selected:
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            else:
                self.master.image_viewer.activate_draw()

```

```

def crop_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.crop_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:

self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()
            else:
                self.master.image_viewer.activate_crop()

def filter_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.filter_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()

        self.master.filter_frame = FilterFrame(master=self.master)
        self.master.filter_frame.grab_set()

def adjust_button_released(self, event):
    self.master.processed_image = self.master.original_image.copy()
    self.master.image_viewer.show_image()
    if self.winfo_containing(event.x_root, event.y_root) ==
self.adjust_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:
                self.master.image_viewer.deactivate_crop()

        self.master.adjust_frame = AdjustFrame(master=self.master)

```

```
self.master.adjust_frame.grab_set()

def clear_button_released(self, event):
    if self.winfo_containing(event.x_root, event.y_root) ==
self.clear_button:
        if self.master.is_image_selected:
            if self.master.is_draw_state:
                self.master.image_viewer.deactivate_draw()
            if self.master.is_crop_state:

self.master.image_viewer.deactivate_crop()

        self.master.processed_image = self.master.original_image.copy()
        self.master.image_viewer.show_image()
```


FILTERFRAME.PY

```
from tkinter import Toplevel, Button, RIGHT
from tkinter import*
import numpy as np
import cv2

class FilterFrame(Toplevel):

    def __init__(self, master=None):
        Toplevel.__init__(self, master=master,bg="pale green")

        self.original_image = self.master.processed_image
        self.filtered_image = None

        self.negative_button = Button(master=self,
text="Negative",width=10,bg="sandy brown")
        self.black_white_button = Button(master=self, text="Black
White",width=10,bg="violet")
        self.sepia_button = Button(master=self,
text="Sepia",width=10,bg="light salmon")
        self.emboss_button = Button(master=self,
text="Emboss",width=10,bg="khaki")
        self.gaussian_blur_button = Button(master=self, text="Gaussian
Blur",width=10,bg="light cyan")
        self.median_blur_button = Button(master=self, text="Median
Blur",width=10,bg="powder blue")
        self.details_button = Button(master=self,
text="Details",width=10,bg="powder blue")
        self.summer_button = Button(master=self,
text="Summer",width=10,bg="powder blue")
        self.winter_button = Button(master=self,
text="Winter",width=10,bg="powder blue")
        self.daylight_button = Button(master=self,
text="DayLight",width=10,bg="powder blue")
        self.grainy_button = Button(master=self,
text="Grainy",width=10,bg="powder blue")
```

```

self.highcontrast_button=Button(master=self,text="High
Contrast",width=10,bg="powder blue")
    self.smoothen_button = Button(master=self,
text="Smoothen",width=10,bg="powder blue")
    self.vignette_button = Button(master=self,
text="Vignette",width=10,bg="powder blue")
    self.bonus_button = Button(master=self,
text="Bonus",width=10,bg="powder blue")
    self.distorted_button = Button(master=self,
text="Distorted",width=10,bg="powder blue")
    self.cancel_button = Button(master=self,
text="Cancel",width=10,bg="powder blue")
    self.apply_button = Button(master=self,
text="Apply",width=10,bg="powder blue")

self.details_button=Button(master=self,text="Details",width=10,bg="powde
r blue")

    self.negative_button.bind("<ButtonRelease>",
self.negative_button_released)
    self.black_white_button.bind("<ButtonRelease>",
self.black_white_released)
    self.sepia_button.bind("<ButtonRelease>", self.sepia_button_released)
    self.emboss_button.bind("<ButtonRelease>",
self.emboss_button_released)
    self.gaussian_blur_button.bind("<ButtonRelease>",
self.gaussian_blur_button_released)
    self.median_blur_button.bind("<ButtonRelease>",
self.median_blur_button_released)
    self.details_button.bind("<ButtonRelease>",
self.details_button_released)
    self.summer_button.bind("<ButtonRelease>",
self.summer_button_released)
    self.winter_button.bind("<ButtonRelease>",
self.winter_button_released)
    self.daylight_button.bind("<ButtonRelease>",
self.daylight_button_released)
    self.grainy_button.bind("<ButtonRelease>",
self.grainy_button_released)

```

```
self.smoothen_button.bind("<ButtonRelease>",
self.smoothen_button_released)
    self.highcontrast_button.bind("<ButtonRelease>",
self.highcontrast_button_released)
    self.distorted_button.bind("<ButtonRelease>",
self.distorted_button_released)
    self.vignette_button.bind("<ButtonRelease>",
self.vignette_button_released)
    self.bonus_button.bind("<ButtonRelease>",
self.bonus_button_released)
    self.apply_button.bind("<ButtonRelease>", self.apply_button_released)
    self.cancel_button.bind("<ButtonRelease>",
self.cancel_button_released)
```

```
self.negative_button.pack()
self.black_white_button.pack()
self.sepia_button.pack()
self.emboss_button.pack()
self.gaussian_blur_button.pack()
self.median_blur_button.pack()
self.details_button.pack()
self.summer_button.pack()
self.winter_button.pack()
self.distorted_button.pack()
self.daylight_button.pack()
self.grainy_button.pack()
self.smoothen_button.pack()
self.highcontrast_button.pack()
self.vignette_button.pack()
self.bonus_button.pack()
self.cancel_button.pack(side=RIGHT)
self.apply_button.pack()
```

```
def details_button_released(self, event):
    self.details()
    self.show_image()
def bonus_button_released(self, event):
    self.bonus()
    self.show_image()
```

```

def distorted_button_released(self, event):
    self.distorted()
    self.show_image()
def highcontrast_button_released(self, event):
    self.highcontrast()
    self.show_image()
def summer_button_released(self, event):
    self.summer()
    self.show_image()
def winter_button_released(self, event):
    self.winter()
    self.show_image()
def daylight_button_released(self, event):
    self.daylight()
    self.show_image()
def grainy_button_released(self, event):
    self.grainy()
    self.show_image()
def smoothen_button_released(self, event):
    self.smoothen()
    self.show_image()
def vignette_button_released(self, event):
    self.vignette()
    self.show_image()

# dst = cv2.stylization(self.original_image, sigma_s=60, sigma_r=0.07)
# dst_gray, dst_color = cv2.pencilSketch(self.original_image,
sigma_s=60, sigma_r=0.07, shade_factor=0.05)

def details(self):
    # sigma_s controls how much the image is smoothed - the larger its
value,
    # the more smoothed the image gets, but it's also slower to compute.
    # sigma_r is important if you want to preserve edges while smoothing
the image.
    # Small sigma_r results in only very similar colors to be averaged (i.e.
smoothed), while colors that differ much will stay intact.
    kernel_sharpening = np.array([[ -1, -1, -1],
                                  [-1, 9, -1],
                                  [-1, -1, -1]])

```

```

[-1, -1, -1]])
    dst2 = cv2.filter2D(self.original_image, -1, kernel_sharpening)
    self.filtered_image=dst2

def gamma_function(self,channel, gamma):
    invGamma = 1 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
                      for i in np.arange(0, 256)]).astype("uint8") # creating
lookup table
    channel = cv2.LUT(channel, table)
    return channel

def summer(self):
    img = self.original_image
    img[:, :, 0] = self.gamma_function(img[:, :, 0], 0.75) # down scaling
blue channel
    img[:, :, 2] = self.gamma_function(img[:, :, 2], 1.25) # up scaling red
channel
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    hsv[:, :, 1] = self.gamma_function(hsv[:, :, 1], 1.2) # up scaling
saturation channel
    img = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    self.filtered_image=img

def winter(self):
    img = self.original_image
    img[:, :, 0] = self.gamma_function(img[:, :, 0], 1.25) # down scaling
blue channel
    img[:, :, 2] = self.gamma_function(img[:, :, 2], 0.75) # up scaling red
channel
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    hsv[:, :, 1] = self.gamma_function(hsv[:, :, 1], 0.8) # up scaling
saturation channel
    img = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    self.filtered_image=img

def vignette(self):
    rows, cols = self.original_image.shape[:2]

```

```

# generating vignette mask using Gaussian
# resultant_kernels
X_resultant_kernel = cv2.getGaussianKernel(cols, 200)
Y_resultant_kernel = cv2.getGaussianKernel(rows, 200)

# generating resultant_kernel matrix
resultant_kernel = Y_resultant_kernel * X_resultant_kernel.T

# creating mask and normalising by using np.linalg
# function
mask = 255 * resultant_kernel / np.linalg.norm(resultant_kernel)
output = np.copy(self.original_image)

# applying the mask to each channel in the input image
for i in range(3):
    output[:, :, i] = output[:, :, i] * mask
self.filtered_image=output

def smoothen(self):
    dst2 = cv2.edgePreservingFilter(self.original_image, flags=1,
sigma_s=60, sigma_r=0.4)
    self.filtered_image = dst2

def bonus(self):
    dst = cv2.stylization(self.original_image, sigma_s=40, sigma_r=0.03)
    self.filtered_image = dst

def distorted(self):
    dst_gray, dst_color = cv2.pencilSketch(self.original_image,
sigma_s=60, sigma_r=0.07, shade_factor=0.05)
    self.filtered_image = dst_color

def daylight(self):
    img = self.original_image
    image_HLS = cv2.cvtColor(img, cv2.COLOR_BGR2HLS) #
Conversion to HLS
    image_HLS = np.array(image_HLS, dtype=np.float64)
    daylight = 1.15

```

```

image_HLS[:, :, 1] = image_HLS[:, :, 1] * daylight # scale pixel values up
for channel 1(Lightness)
    image_HLS[:, :, 1][image_HLS[:, :, 1] > 255] = 255 # Sets all values
above 255 to 255
    image_HLS = np.array(image_HLS, dtype=np.uint8)
    image_RGB = cv2.cvtColor(image_HLS, cv2.COLOR_HLS2BGR)
    self.filtered_image=image_RGB

```

```

def grainy(self):
    img = self.original_image
    height, width = img.shape[:2]
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    thresh = 0.8 # creating threshold. This means noise will be added to
80% pixels

```

```

    for i in range(height):
        for j in range(width):
            if np.random.rand() <= thresh:
                if np.random.randint(2) == 0:
                    gray[i, j] = min(gray[i, j] + np.random.randint(0, 64),
255) # adding random value between 0 to 64.

```

Anything above 255 is set to 255.

```

        else:
            gray[i, j] = max(gray[i, j] - np.random.randint(0, 64),
0) # subtracting random values between 0 to 64.

```

Anything below 0 is set to 0.

```

        self.filtered_image=gray

```

```

def highcontrast(self):
    img = self.original_image
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    xp = [0, 64, 112, 128, 144, 192, 255] # setting reference values
    fp = [0, 16, 64, 128, 192, 240, 255] # setting values to be taken for
reference values
    x = np.arange(256)
    table = np.interp(x, xp, fp).astype('uint8') # creating lookup table
    img = cv2.LUT(gray, table) # changing values based on lookup table
    self.filtered_image = img

```

```

def negative_button_released(self, event):

```

```

self.negative()
    self.show_image()

def black_white_released(self, event):
    self.black_white()
    self.show_image()

def sepia_button_released(self, event):
    self.sepia()
    self.show_image()

def emboss_button_released(self, event):
    self.emboss()
    self.show_image()

def gaussian_blur_button_released(self, event):
    self.gaussian_blur()
    self.show_image()

def median_blur_button_released(self, event):
    self.gaussian_blur()
    self.show_image()

def apply_button_released(self, event):
    self.master.processed_image = self.filtered_image
    self.show_image()
    self.close()

def cancel_button_released(self, event):
    self.master.image_viewer.show_image()
    self.close()

def show_image(self):
    self.master.image_viewer.show_image(img=self.filtered_image)

def negative(self):
    self.filtered_image = cv2.bitwise_not(self.original_image)

def black_white(self):

```



```

self.filtered_image = cv2.cvtColor(self.original_image,
cv2.COLOR_BGR2GRAY)
    self.filtered_image = cv2.cvtColor(self.filtered_image,
cv2.COLOR_GRAY2BGR)

def sepia(self):
    kernel = np.array([[0.272, 0.534, 0.131],
                      [0.349, 0.686, 0.168],
                      [0.393, 0.769, 0.189]])

    self.filtered_image = cv2.filter2D(self.original_image, -1, kernel)

def emboss(self):
    kernel = np.array([[0, -1, -1],
                      [1, 0, -1],
                      [1, 1, 0]])

    self.filtered_image = cv2.filter2D(self.original_image, -1, kernel)

def gaussian_blur(self):
    self.filtered_image = cv2.GaussianBlur(self.original_image, (41, 41), 0)

def median_blur(self):
    self.filtered_image = cv2.medianBlur(self.original_image, 41)

def close(self):
    self.destroy()

```

INIT.PY

```

from main import Main

```

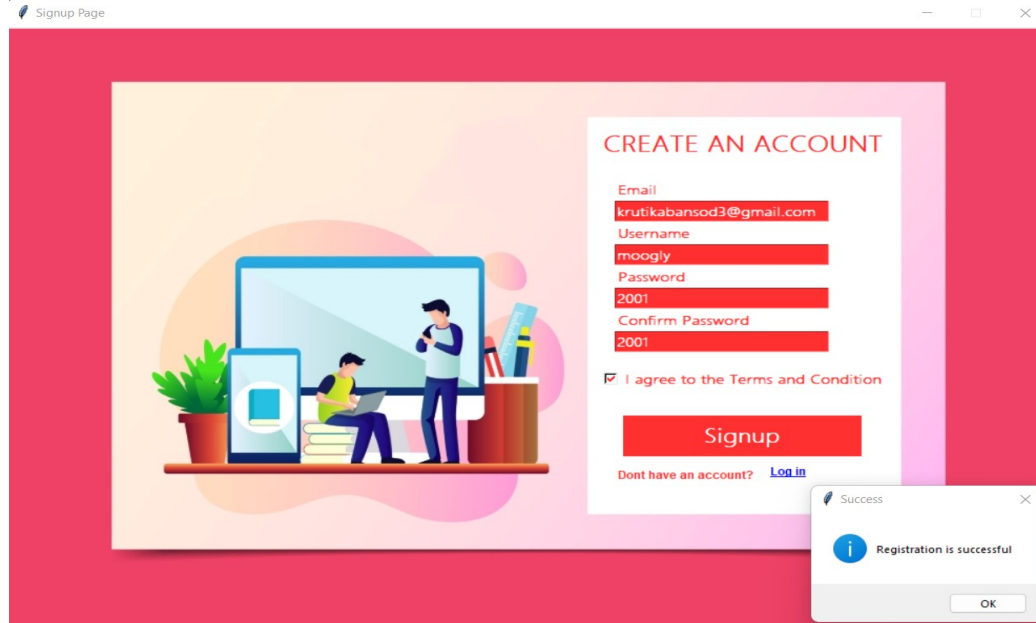
```

root = Main()
root.mainloop()

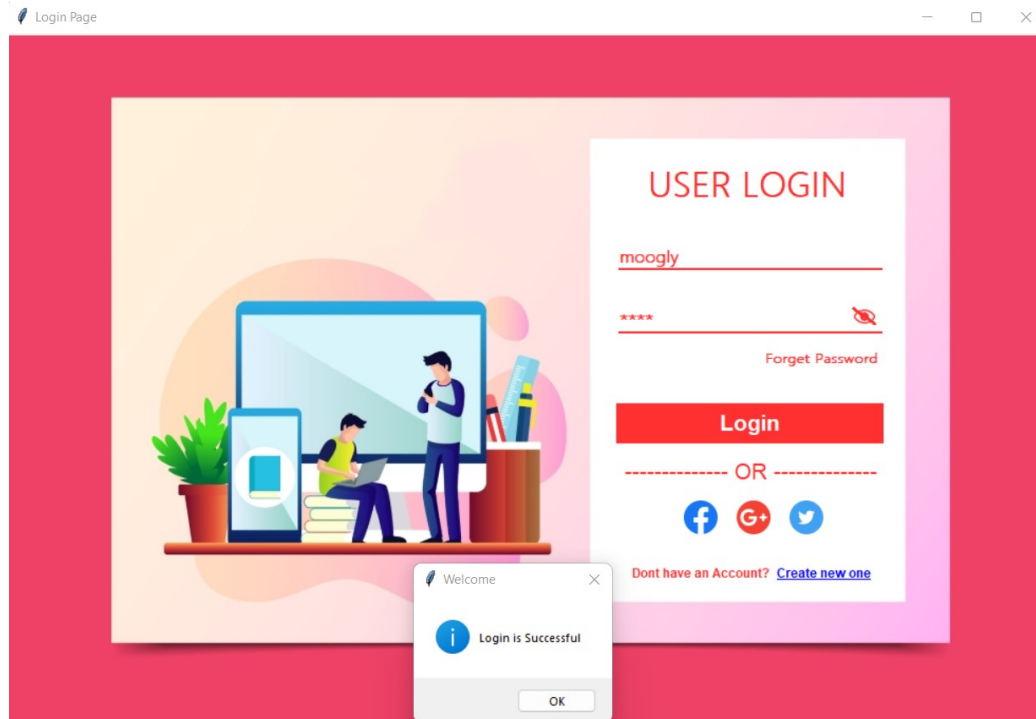
```

OUTPUT SCREEN

REGISTRATION FORM



LOGIN FORM



CHANGE PASSWORD

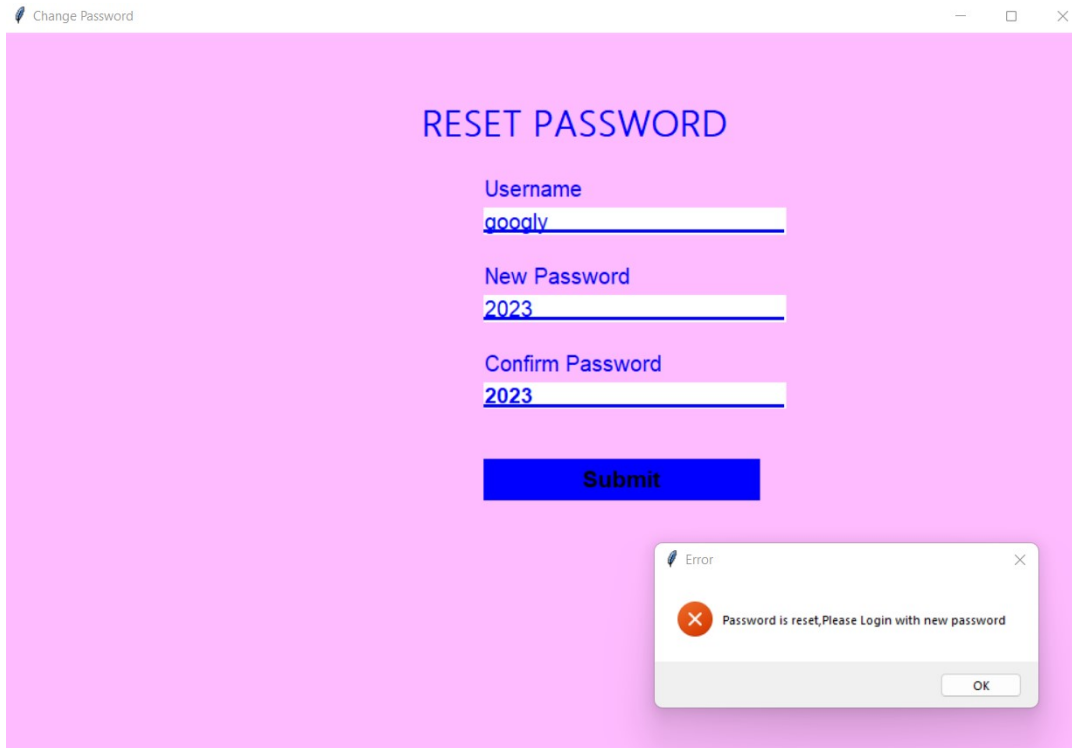
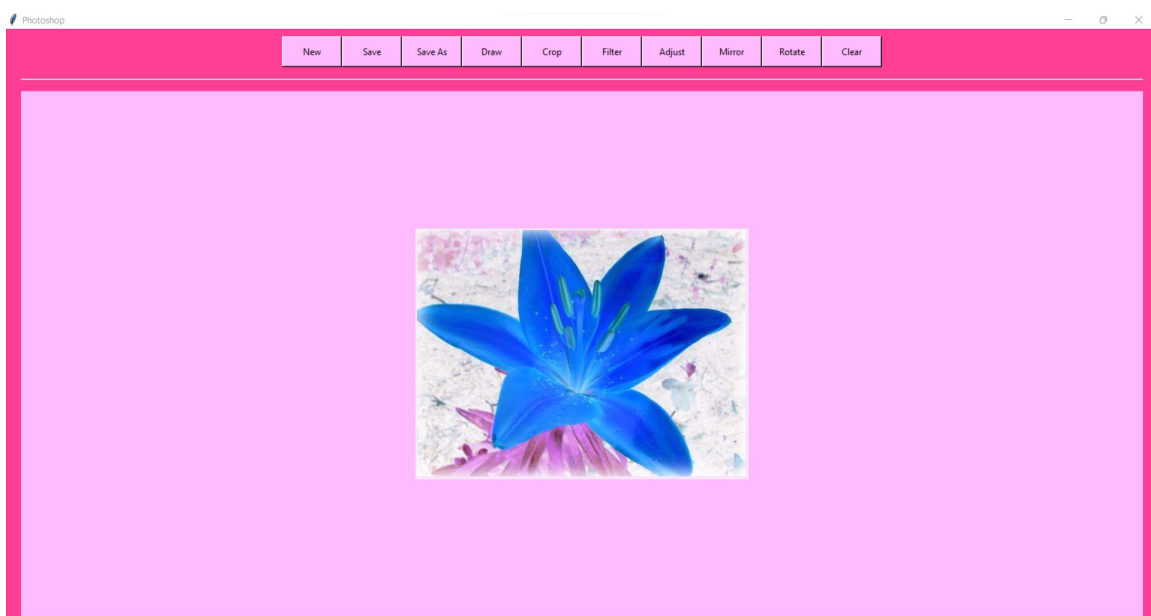


IMAGE EDITING



Testing And Validation Checks

Testing

Testing plays important role to identify the quality of any software. Testing actually refers to detecting errors in the system. Before testing can begin, a test plan needs to be developed. Test plan actually includes the type of testing that has to be performed on the code, resources for testing, how the software will be tested. There are several types of testing during the test phase, that includes quality assurance testing (QAT), System Integration testing (SIT), and user acceptance testing (UAT).

- Quality Assurance (QA) Testing: In this the procedures and processes are checked. This means whether the instructions are executed as per the user requirements and commands.
- System Integration Testing (SIT): It verifies proper execution of software components and proper interfacing between components within the solution. the objective behind this testing is to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution.
- User Acceptance Testing (UAT): This is the last phase of the software testing procedure. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to actual use.

Before testing the system, we need to consider following questions in our mind:

- What is the actual problem?

- How critical the problem is?
- Measures should be taken for the upcoming problems or errors?

Testing gives chance to upgrade or to improve if any drawbacks prevails in the application. Testing is generally done at two levels, testing of individual modules and testing entire system.

During system testing, the system is used experimentally to ensure that the software does not fall. that it will run according to its specification and in the way users expect. Testing is done throughout system development at various stages.

Following are the type of testing done in the project:

1. Program Testing: In this, we have to concentrate on the software part, system software should be free from errors. whether it is syntax error or logical error. In this system, we have done software testing and the output of this test is satisfactory. It fulfills all the conditions, which was required for the program testing.
2. Stress Testing: this test is conducted to check the performance of the system in main hours. It finds out how much workload the system can bear. In stress testing of this system, we come to know that this software can work easily and accurately at any condition. The concentration is made on the performance of the system by checking the giving input and their expected outputs.
3. Documentation Testing: this testing work to find out that whatever document supplied is satisfactory or there is

need to supply further document. In this system, all the documents which are supplied are satisfactory.

Different types of Testing

1) Unit Testing

Unit testing is a type of software testing which is done on an individual unit or component to test its correctness. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Unit, JUnit for the test execution. Unit testing is important because we can find more defects at the unit test level.

For example, there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

a) White Box Testing

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

b) Gorilla Testing

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

2) Integration Testing

Integration testing is a type of software testing where two or more modules of an application is logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow

among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system. This type of testing is done on integrating modules of a system or between systems.

For example, a user is buying a flight ticket from any airline website.

Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

a) gray box testing

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

3) System Testing

System testing is types of testing where tester evaluates the whole system against the specified requirements.

a) End to End Testing

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.

b) Black Box Testing

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test.

Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found [here](#).

c) Smoke Testing

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

4) Acceptance Testing

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected, this is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

a) Alpha Testing

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

b) Beta Testing

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the Real Environment before releasing the product to the market for the actual end-users. Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Non-Functional Testing

There are four main types of functional testing.

1) Security Testing

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.

2) Performance Testing

Performance testing is testing of an application's stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users.

Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner,

etc. are good tools available in the market.

3) Usability Testing

Usability testing is testing an application from the user's perspective to check the look and feel and user-friendliness.

For example, there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background color of the app should be black and price of and stock is displayed in red or green color.

4) Compatibility testing

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment. Compatibility testing ensures that software can run on different configuration, different databases, different browsers, and their versions. The testing team performs compatibility testing.

Other Types of Testing

1. Back-end Testing

Whenever an input or data is entered on the front-end application, it is stored in the

database and the testing of such database is known as Database Testing or Backend Testing.

There are different databases like SQL Server, MySQL, Oracle, etc.

Database Testing involves testing of table structure, schema, stored procedure, data structure, and so on. In Back-end Testing, GUI is not involved, the testers are directly connected to the database with proper access and testers can easily verify data by running a few queries on the database.

2.Black Box Testing

Internal system design is not considered in this type of testing. Tests are based on the requirements and functionality. Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

3. Browser Compatibility Testing

This is a sub-type of Compatibility Testing (which is explained below) and is performed by the testing team.

Browser Compatibility Testing is performed for web applications and ensures that the software can run with a combination of different browsers and operating systems.

This type of testing also validates whether a web application runs on all versions of all browsers or not.

4. Backward Compatibility Testing

It is a type of testing that validates whether the newly developed software or updated software works well with the older version of the environment or not.

Backward Compatibility Testing checks whether the new version of the software works properly with the file format created by an older version of the software. It

also works well with data tables, data files, and data structures created by the older

version of that software. If any of the software is updated, then it should work well on top of the previous version of that software.

VALIDATION CHECKS

Data validation is the process of ensuring, at least as far as is possible, that the data given to a program by a user or from a file (essentially, the system's input) is of the correct type, and in the correct format. There are however measures that can be taken to restrict the program's input to valid data. Such measures involve the application of validation rules to any data being input to the program. In this system, Data validation rules can also make this system more user friendly, since they enable the program to warn the user immediately when there is a problem rather than simply allowing them to continue entering data until the program crashes or some other problem occurs.

In this proposed system, we have introduced the following data validation rules:

1. Value entered check: this is used for things like required fields in online forms where the user must enter some data (for example their username and password) and must not leave the field blank.
2. Permitted character check: it is useful for determining whether an input string contains valid characters. For example, a phone number may include the digits 09.
3. Limit check: It is used for numeric values that must either be greater than or equal to some lower limit, or less than or equal to some upper limit. For example, the limited number that a user can enter as a phone number is

4. Confirmation check: At the time of creating an account in this system, it is used for determining whether the enter password and confirm password are same or not.

5. Email address check: At the time of creating an account, the system can only accept a valid email id. For example, “@gmail.com”.

System Security Measures

The objective of system security is the protection of information and property from theft, corruption and other types of damage, while allowing the information and property to remain accessible and productive. System security includes the development and implementation of security countermeasures. There are a number of different approaches to system security.

In this proposed system, we have provided the following security majors:

1. Password: the most widely method to prevent unauthorized access is to use passwords. The password needs to be kept secret and is only intended for the specific user. In this system, each password is associated with a specific username since many individuals may be accessing the same system.

2. Accessibility: In the game, with the help of admin username and password, only the admin has the right to update the game such as add, delete and update the products.

- **Physical:**

The sites containing computer systems must be physically secured against armed and malicious intruders. The workstations must be carefully protected.

- **Human:**

Only appropriate users must have the authorization to access the system. Phishing(collecting confidential information) and Dumpster

Diving(collecting basic information so as to gain unauthorized access) must be avoided.

- **Operating system:**

The system must protect itself from accidental or purposeful security breaches.

- **Networking System:**

Almost all of the information is shared between different systems via a network. Intercepting these data could be just as harmful as breaking into a computer. Henceforth, Network should be properly secured against such attacks.

Security System Goals

1. **Integrity:** The objects in the system mustn't be accessed by any unauthorized user & any user not having sufficient rights should not be allowed to modify the important system files and resources.
2. **Secrecy:** The objects of the system must be accessible only to a limited number of authorized users. Not everyone should be able to view the system files.
3. **Availability:** All the resources of the system must be accessible to all the authorized users i.e only one user/process should not have the right to hog all the system resources. If such kind of situation occurs, denial of service could happen. In this kind of situation, malware might hog the resources for itself & thus preventing the legitimate processes from accessing the system resources.

Implementation, Evaluation, Maintenance

Implementation

Implementation refers to that stage of project during which the theory is turned into practice i.e. converting soft ideas into actual process. In this stage physical system specifications are converted into working and reliable solution. This is where the system is developed. It is followed by testing and then again implementation.

Implementation phases:

- **Coding:** this includes implementation of the design document into executable programming language code. The output of the coding phase is the source code for the software that acts as input to the testing and maintenance phase.
- **Integration and Testing:** It includes detection of errors in the software. The testing process starts with a test plan recognizes test-related activities , such as test case generation, testing criteria and resource allocation of testing. The code is tested and mapped against the design document created in the design phase.
- **Installation:** New system is installed and rolled out.

The steps involved in this phase are:

1. Acquisition and installation of hardware and software.

2. Conversion: It actually means to convert the old data to new format for proper functioning of the application in the new system.
3. User Training: User in this case has to be trained to use the system properly so that it is easy for them to grab control over the use of the application.
4. Documentation: This provides details of how to operate the given software, application and website.

The hardware and relevant software required for running the application must be installed and fully checked before implementation. In this phase conversion plays a crucial role. It actually means to convert the old data to a new format for proper functioning of the application in the new system.

During the phase all the required programs are loaded onto user's computer. User must be trained.

The documentation is a complete description of the system from the users point of view as it provides details of how to operate the given software and application. It also includes certain error messages that a user is expected to encounter during its usage and solution to the expected problems. It involves detailed and step by step information of the project development so as to modify or update as per the new user requirements.

Evaluation

Evaluation is a strategy used to determine the success and impact of projects, programs, or policies. It requires the evaluator to gather important information to analyze the process and outcome of a certain project. Project evaluation prompts changes in internal workflow, detects patterns in the target audience of the project, plans for upcoming projects or reports the value of projects to external stakeholders.

Types of project evaluation

The following are common types of project evaluation to implement in your projects:

Pre-project evaluation

Before beginning a project, your team could evaluate whether it is feasible to complete successfully. This often takes place naturally in the developmental stage of projects and is crucial for the effective execution of the project. It is important that all involved are aware of the objectives and goals before work begins.

Ongoing evaluation

Throughout the life cycle of the project, you may use metrics to verify completed tasks. This includes budget, percentage of completed tasks and the overall quality of the work delivered so far. Try to remain focused on your original objectives and goals as the project is underway, so your team remains on track.

Post-project evaluation

After the project is complete, it is important to analyze the outcomes and impacts of the project. Outcomes help measure how effective the project was in meeting the objectives and goals set at the beginning. Impacts may determine how successful the project was in creating a tangible change for the target audience.

Self-evaluation

At any point in the life cycle of the project, an individual can conduct a self-evaluation. Self-evaluation analyzes if their work is contributing to greater objectives and goals. Recognizing strengths and weaknesses, measuring their successes, and determining the scope of their impact can increase their ability to work effectively as part of the team.

External evaluation

Another option is hiring external agencies to perform evaluations for your projects. These agencies typically have no prior connection or involvement in the project, leading to a high level of impartiality when conducting the evaluation and concluding. External evaluation is valuable for projects that include a large number of stakeholders or have several moving pieces.

Maintenance

Maintenance is the final stage after the development process. After the system is installed, it must be maintained means that the computer programs must be modified and kept up to date. The average amount of time spent on maintenance is 60% of the total time. Estimates of the time spent by departments on maintenance have ranged from 48 to 60 percent of the total time spent developing systems. As the number of programs written increases, so does the amount of maintenance they require.

Maintenance covers a wide range of activities including correcting, coding, designing errors and updating user support. the project needs maintenance in further if any enhancements are made, maintenance of the hardware and software is also required.

The maintenance phase occurs once the system is operational. It includes implementation of changes that software might undergo over a period of time, or implementation of new requirements after the software is deployed at the customer location. The maintenance phase also includes handling the residual errors that may exist in the software even the testing phase.

The maintenance phase also monitors system performance, rectifies bugs and requested changes are made. Maintenance is performed for two reasons:

- First is to correct software errors. No matter how thoroughly the system is tested, bugs or errors creep into the computer programs. Bugs in

commercial. PC software are often documented as “known anomalies” and are corrected when new versions of the software are released or in an interim release. In custom software (also called bespoke software), bugs must be corrected as they are detected.

- Second, for performing system maintenance is to enhance the software’s capabilities in response to changing organizational needs, generally involving one of the following three situations:

1. Users often request additional features after they become familiar with the computer system and its capabilities.

2. Hardware and software are changing at an accelerated pace. In summary, Maintenance is an ongoing process over the life cycle of a system. After the application is installed, maintenance usually takes the form of correcting previously undetected program errors.

3. Once these are corrected, the system approaches a steady state, providing dependable service to its users.

All the activities that occur after the completion of the program come under the program maintenance. Moreover, it is not a part of the implementation process directly but, it plays an important role. Furthermore, the cost of maintenance is more than the cost of development. Functions of program maintenance include the following:

- Finding and correcting the errors.
- Modifying the program to enhance it. Besides, we modify it usually to adapt to some new laws or government policies. Moreover, we also have to modify it in case if we want to change the hardware or operating system.

- Update the documentation part.
- Add new features and functions.
- Remove useless parts of code. Types of Program Maintenance

The types are as follows:

Corrective Maintenance

In this process, the developers fix the errors which come after the use of the software.

Moreover, the users themselves point out these errors.

Preventive Maintenance

This acts as a precaution since we modify the code to avoid any errors in the future.

Adaptive Maintenance

The changes in our environment may sometimes require making changes in the software as well. Therefore, it is adaptive maintenance. For example, changing the software according to some new rules of the government.

Perfective Maintenance

This involves aiming at keeping the software up to date with the latest technology. Moreover, programmers include any recommendations from the users, in the software.

Future Scope of Project

- **Extra Features:** Success of this application provides the extra ordinary features to the user.
- **Simple and Easy:** This project procedure is user-friendly, easy, simple and error free, it makes the software attractive.
- **Reliability:** This application can be run or expected into the current operating software's also.
- **Platform:** Image Editing Application will arcade- style in editing platform.
- **Use:** The application use as an environments.
- **Vision:** The vision of the project, in terms of understanding scope, requires detailed foresight about what they will be its scope, requires of hours of editing photo, graphics quality, number of filter supported, quality of feature, and other aspects that would be thought out fully in a design images.
- We will be regularly updating our application will try to add and make it more informational and creative.
- We will try to make it more user friendly so it can ne easily accessible.
- We will be adding more sections in our application.
- We will increase compatibility of our application so it can be used both online or offline.

Conclusion

- ❖ Premium Quality: The main motive behind developing this project is to provide good quality software.
- ❖ Fulfilment of User Satisfaction: The basic concern behind developing this software is to provide good quality of and provide high level of satisfaction to the user.
- ❖ We have created this application using python language which is very popular.
- ❖ Our application also has many pros and cons which will be improved as per the user's preferences.
- ❖ This application is all about improvement and 100% beneficial.

Bibliography And References

References

1. Mike McGrath, McGraw-Hill, New Delhi
2. Laura cassell and Alan Gauld
3. Mark Lutz and David Beazley
4. <http://.tutorialspoint.com>
5. YouTube

A
PROJECT SYNOPSIS
ON

“PHOTO SHOP AND EDITING APPLICATIONS”

Submitted to

G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS
In the Partial Fulfillment of

B.Com. (Computer Application) Final Year

Synopsis Submitted by

Krutika Bansod
Palak Yadav

Under the Guidance of

Pravin J. Yadao



Shiksha Mandal's

G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS
2022-2023

1. Introduction

The project aims at developing a image editor using python .Image editor is a small application which lets the user to manipulate digital image using tools such as color adjustments, sharpening, zoom, rotation, print etc. Efficient GUI has been provided which let the user to access this tools Editor access the image in arrays of pixels. These pixels contain the image color and brightness information. Image editor can change the pixels to enhance the image in many ways. The pixels can be changed as a group, or individually, by the algorithms within the image editors and image filter.

2. Objectives of the project:

- The main objective of Image Editor is their users, by making user interaction image-based and context-aware.
- Moreover, users can simply navigate to new, relevant images and texts in the knowledge repository only by clicking on interesting image parts.
- The purpose of Image editing is generally to improve communication with some target groups in mind.
- Image editing is the manipulation and arrangement of Image shots.
- Its to make an interpretation of the clips to tell your story coherently.
Best wishes.

3. Project Category: Application

4. Tools/ Platform/ Languages to be used: Python, Microsoft Access, Visual Studio Code

5. Scope of future application:

- Remove unwanted footage: This is the simplest and most common task in editing. Many Image can be dramatically improved by simply getting rid of the flawed or unwanted bits.
- Create a flow: Most Image server a purpose such as telling a story or providing information. Editing is a crucial step in making sure the Image flows in a way which achieves this goal.
- Give the Image a particular “Angular”: Image can be tailored to support a particular viewpoint, impart a message or serve an agenda.

Submitted by:

**Krutika Bansod
Palak Yadav**

Approved by,

**Prof. Pravin Yadao
Project Guide**