NAVKIS EDUCATIONAL CENTRE

Online Store

Computer Project File

SUBMITTED BY -

NAME: Amar Vaid

CLASS: 12 - A

ACADEMIC YEAR: 2022-23

CERTIFICATE OF COMPLETION

This is to certify that **Amar Vaid**, a student of Computer Science, Navkis Educational Centre, has undergone the completion of Project Synopsis documentation prescribed by the Central Board of Secondary Education (CBSE) for the All India Senior School Certificate Examination (AISSCE) for the academic year 2022-2023.

SIGNATURE Of PRINCIPAL : ______________________________

SIGNATURE Of INTERNAL EXAMINER : __________________

SIGNATURE Of EXTERNAL EXAMINER : __________________

REGISTRATION NUMBER : ______________________________

DATE OF SUBMISSION : _____________________________

ACKNOWLEDGEMENT

I am deeply thankful to my school, **Navkis Educational Centre** and Principal, **Mrs. Seema Gupta,** who has given me this opportunity and encouragement to complete this Project Synopsis documentation.

I take this opportunity to acknowledge our Computer faculty **Mrs. Karthika V** for providing me with valuable support, guidance and advice on planning and compilation of the Project Synopsis.

I would like to thank my parents and friends for the smooth completion and documentation of the Project Synopsis for the **academic session 2022-23.**

This Computer Project Synopsis based on Python programming language has been documented by **Amar Vaid**

NAME	: ______________________________
SIGNATURE	: ______________________________
DATE	: ______________________________
PLACE	: ______________________________

INDEX

INTRODUCTION

Shopping is a necessity to get by in our daily life but it also serves as a form of entertainment for some to see the latest things in the market.

While many enjoy this experience it is not always feasible for everyone to visit stores and shop. This is where online shopping has come in handy in the past decade making shopping easier now than it was ever before by remotely connecting the customer directly to the seller making the process easier for both parties.

Everything from cars to simple daily requirements such as food can be ordered online these days. This project aims to create a program for the same and make the experience immersive yet easy to access.

OBJECTIVE And Scope Of Project

The objective of this project is to create an interactive interface for a user where they can access information about all the products/accessories and provide various features and categories to make things easier for the user.

To create a one-shop stop for all users allowing them access to all information about all the products directly from the seller as well as purchase items

It also aims to satisfy all users' needs, customers and sellers alike by providing them with a clean and interactive interface to purchase and sell items.

Given enough time and resources, we can not only introduce many more features but we can also add more data and information about the product including pictures of the product all while maintaining a clean interface for the user to search through different methods such as categories and search bar.

Another way to expand the project would be to add more categories and data in the application making things easier for the user.

PROBLEM DEFINITION AND ANALYSIS

Due to expansion of the online shopping, most retailers and consumers avoid in-person shopping and are switching to online markets. This is mainly due to the maximum number of people opting for online shopping which is much more convenient. For the benefit of their business, so are the retailers switching to online methods. The idea of shopping from the comfort of your home can be quite attractive to some while challenging to others. The driving problem is the management of online stores is much more challenging to many individuals but can be tackled by certain methods such as:

- Creating software that is simple, convenient, and easy to use.
- The software should have all the required features along with being efficient.
- The working of the system should be as efficient as possible with straightforward codes that are easy to execute.
- Along with that, the system should look attractive yet simple.

SYSTEM IMPLEMENTATION

Recommended System Requirements Processors:

- Intel® Core™ i3 processor 4300M at 2.60 GHz.
- Disk space: 2 to 4 GB.
- Operating systems: Windows® 10, MACOS, and UBUNTU.
- Python Versions: 3.X.X or Higher.

Minimum System Requirements Processors:

- Intel Atom® processor or Intel® Core™ i3 processor.
- Disk space: 1 GB.
- Operating systems: Windows 7 or later, MACOS, and UBUNTU.
- Python Versions: 2.7.X, 3.6.X.

Prerequisites before installing MySQL Connector Python

You need root or administrator privileges to perform the installation process. Python must be installed on your machine.

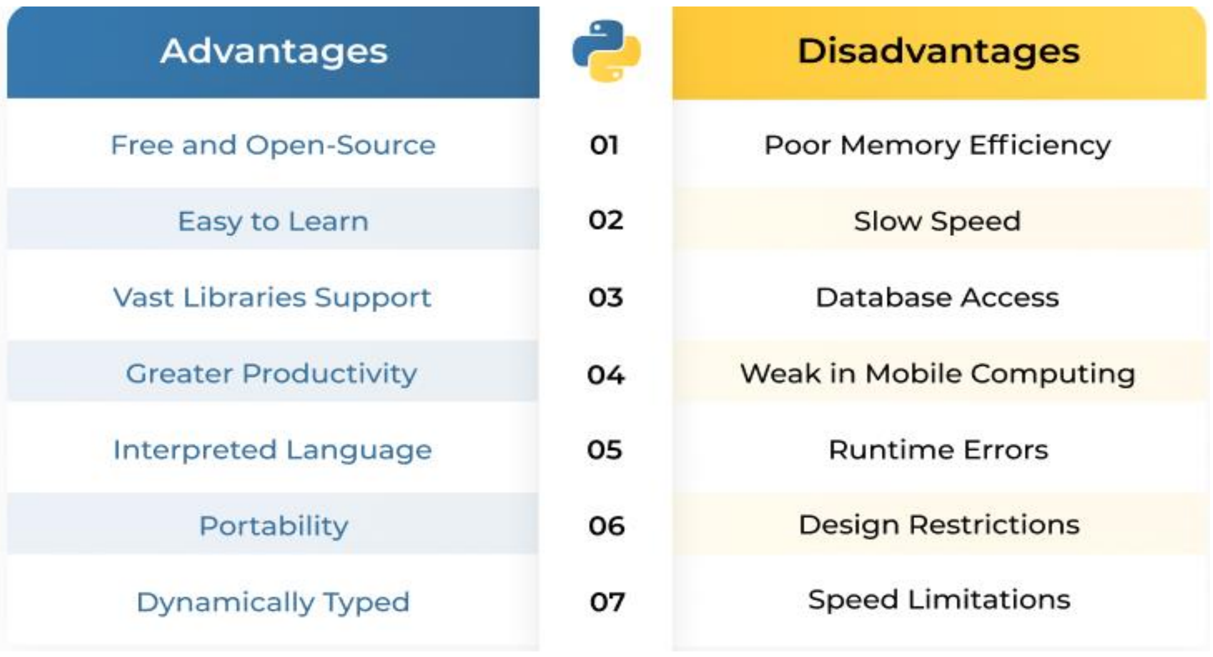
Note: – MySQL Connector Python requires python to be in the system's PATH. Installation fails if it doesn't find Python. On Windows, If Python doesn't exist in the system's PATH, please manually add the directory containing python.exe yourself.

WORKING ENVIRONMENT

Python Overview

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development and for use as a scripting or glue language to connect existing components.
- Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse.

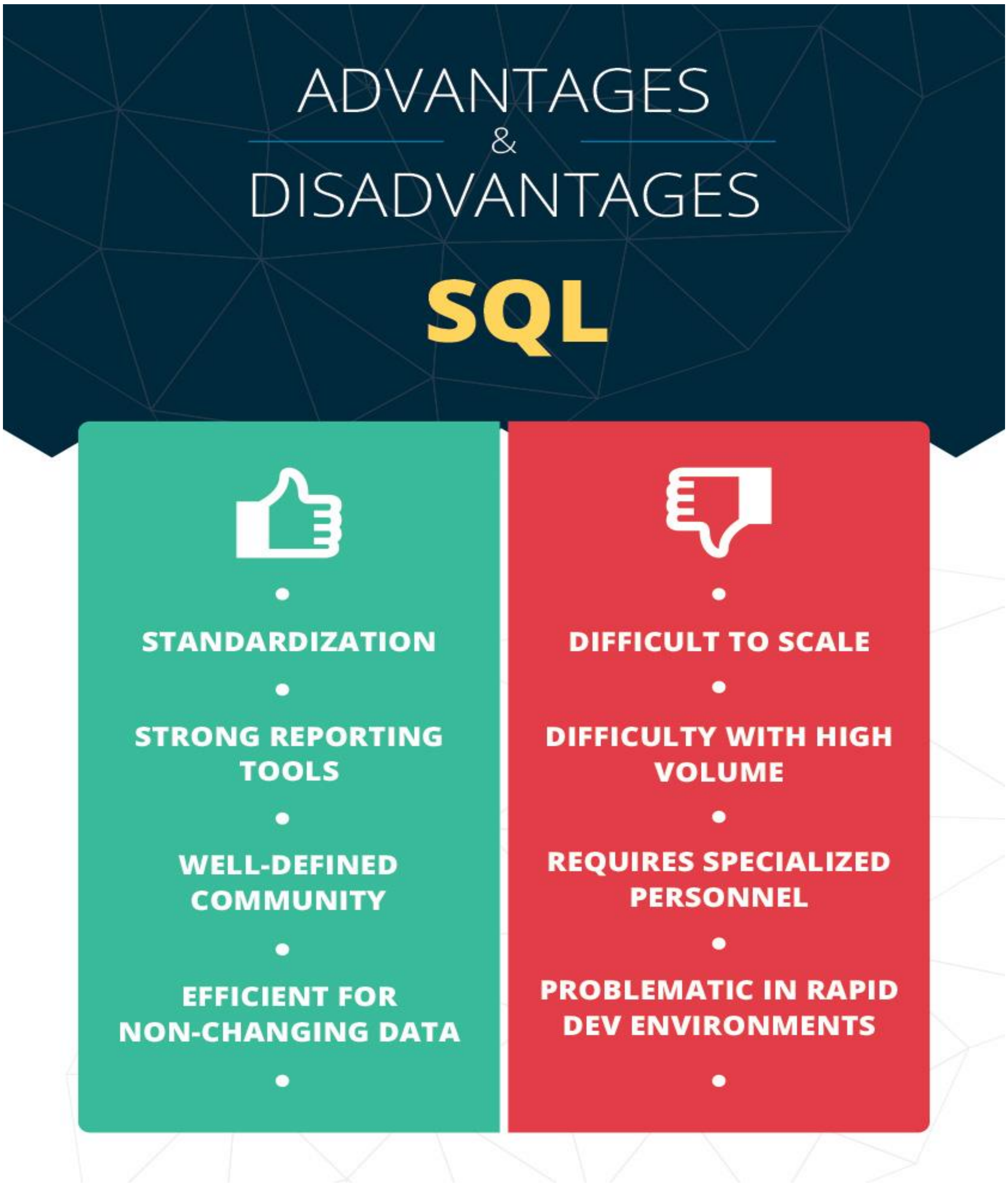
The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.



Advantages		Disadvantages
Free and Open-Source	01	Poor Memory Efficiency
Easy to Learn	02	Slow Speed
Vast Libraries Support	03	Database Access
Greater Productivity	04	Weak in Mobile Computing
Interpreted Language	05	Runtime Errors
Portability	06	Design Restrictions
Dynamically Typed	07	Speed Limitations

MySQL Overview

Structured Query Language, abbreviated as SQL, is a domain-specific language used in programming and designed for managing data held in a relational database management system or stream processing in a relational data stream management system.


ADVANTAGES
&
DISADVANTAGES
SQL
STANDARDIZATION
STRONG REPORTING TOOLS
WELL-DEFINED COMMUNITY
EFFICIENT FOR NON-CHANGING DATA
DIFFICULT TO SCALE
DIFFICULTY WITH HIGH VOLUME
REQUIRES SPECIALIZED PERSONNEL
PROBLEMATIC IN RAPID DEV ENVIRONMENTS

Software Development Life Cycle

Planning

During this first phase of the development life cycle, security considerations are key to diligent and early integration, thereby ensuring that threats, requirements, and potential constraints in functionality and integration are considered. At this point, security is looked at more in terms of business risks with input from the information security office. For example, an agency may identify a political risk resulting from a prominent website being modified or made unavailable during a critical business period, resulting in decreased trust by citizens. Key security activities for this phase include:

- Initial delineation of business requirements in terms of confidentiality, integrity, and availability;
- Determination of information categorization and identification of known special handling requirements to transmit, store, or create information such as personally identifiable information; and Determination of any privacy requirements.

Analysis

This section addresses security considerations unique to the second SDLC phase. Key security activities for this phase include:

- Conduct the risk assessment and use the results to supplement the baseline security controls;
- Analyze security requirements; Perform functional and security testing;
- Prepare initial documents for system certification and accreditation;
- Although this section presents the information security components in a sequential top-down manner, the order of completion is not necessarily fixed. Security analysis of complex systems will need to be iterated until consistency and completeness is achieved.

Design

During this phase of SDLC, the security architecture is designed.

Implementation

During this phase, the system will be installed and evaluated in the organization’s operational environment. Key security activities for this phase include:

- Integrate the information system into its environment;
- Plan and conduct system certification activities in synchronization with the testing of security controls; and
- Complete system accreditation activities.

Coding and Testing

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

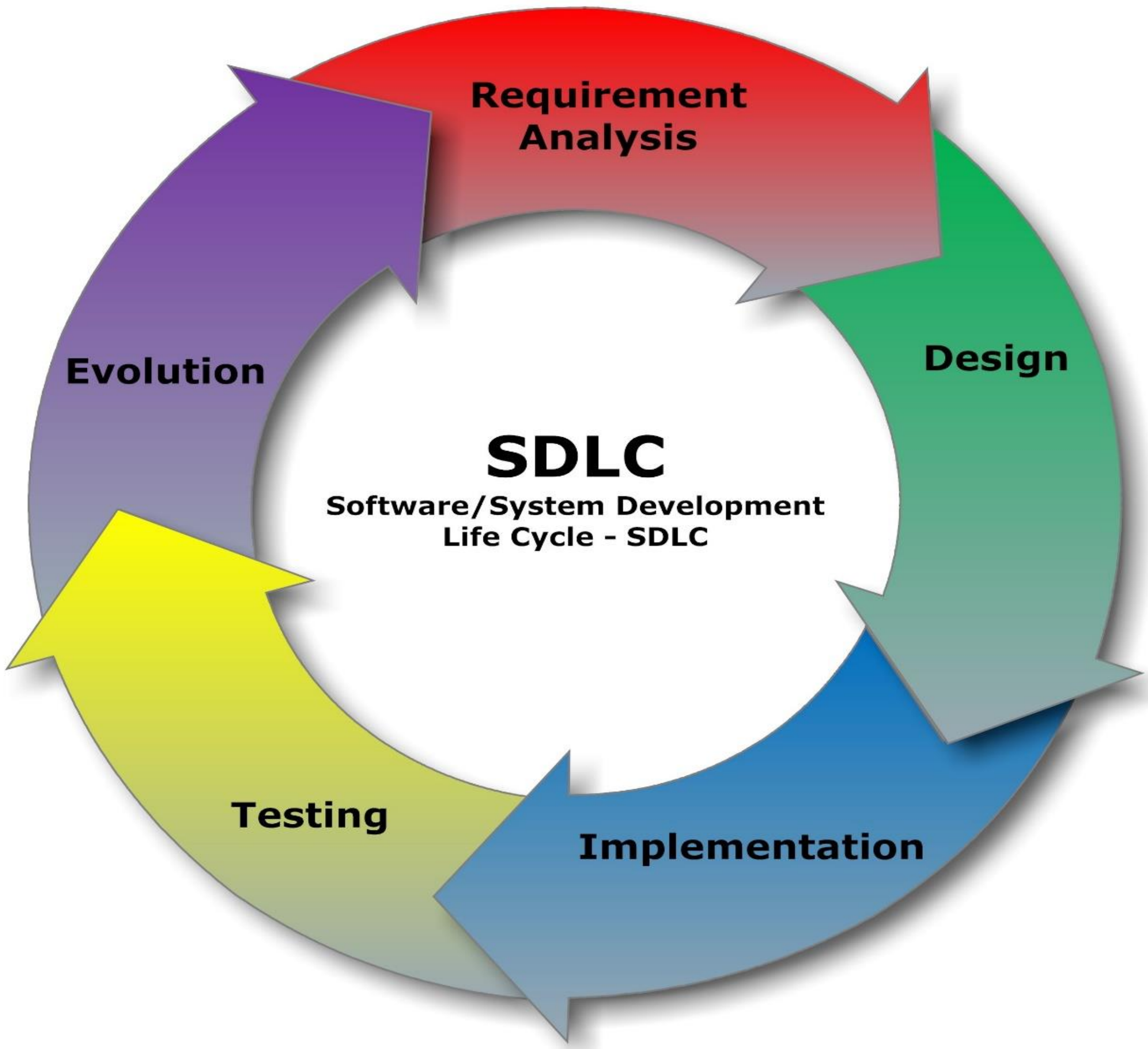
Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, the testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send it back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Maintenance/Support

In this phase, systems are in place and operating, enhancements and/or modifications to the system are developed and tested, and hardware and/or software is added or replaced. The system is monitored for continued performance in accordance with security 13 requirements and needed system modifications are incorporated. The operational system is periodically assessed to determine how the system can be made more effective, secure, and efficient. Operations continue as long as the system can be effectively adapted to respond to an organization’s needs while maintaining an agreed-upon risk level. When necessary modifications or changes are identified, the system may reenter a previous phase of the SDLC. Key security activities for this phase include:

- Conduct an operational readiness review;
- Manage the configuration of the system ;
- Institute processes and procedures for assured operations and continuous monitoring of the information system’s security controls
- Perform reauthorization as required


Requirement Analysis
Design
Implementation
Testing
Evolution
SDLC
Software/System Development Life Cycle - SDLC

Sample Code

Start Screen

```
from tkinter import *


Start = Tk()
Start.configure(bg='SlateBlue2')
Start.geometry('1000x500')


def selling():
    Start.destroy()
    import S_Login


def buying():
    Start.destroy()
    import B_Login
title = Label(Start, text="ONLINE STORE",font=("DejaVu",24),width=40,height=5,bg='SlateBlue2',fg='gold')
title.place(x=120,y=10)


Selling = Button(Start, text= "Seller", command= selling,fg= "white",bg="black", width= 30,height=10,font='STIX')
Selling.place(x=100,y=150)
Buyer = Button(Start, text= "Buyer", command= buying,fg= "white",bg="black", width= 30,height=10,font='STIX')
Buyer.place(x=550,y=150)
Start.mainloop()
```

S_login(Seller Login/Signup)

```
import mysql.connector as sql
from tkinter import *
from PIL import Image,ImageTk
import os
import sys

if sys.argv:
    filepath = sys.argv[0]
    folder, filename = os.path.split(filepath)
    os.chdir(folder)



win = Tk()

win.configure(bg='DarkGoldenRod2')

win.geometry("1000x500")

win.title("Login Page")



def login() :

    db = sql.connect(host = "localhost", user = "root", passwd = "778240")
    cur = db.cursor()



    try :

        cur.execute("create database login")

        db = sql.connect(host = "localhost", user = "root", passwd = "778240", 
database = "login")

        cur = db.cursor()



    except sql.errors.DatabaseError:

        db = sql.connect(host = "localhost", user = "root", passwd = "778240", 
database = "login")

        cur = db.cursor()



        try :

            cur.execute("create table main(username varchar(50), NOT NULL, 
password int NOT NULL)")



        except sql.errors.ProgrammingError:
            pass

    finally :

        try :

            cur.execute("create table main(username varchar(50) NOT NULL, "

                        "password int NOT NULL)")

        except sql.errors.ProgrammingError:

            pass

    while True :

        user = user1.get()

        passwd = passwd1.get()

        cur.execute("select * from main where username = '%s' and password = %s"
% (user, passwd))

        rud = cur.fetchall()

        if rud:
            print("Welcome")

            break


        else:

            cur.execute("insert into main values('{}', {})".format(str(user), passwd))

            db.commit()

            print("Account Created")

            break

    cur.close()
    db.close()
    win.destroy()
    import selling

Title = Label(win, text = "Login",bg='DarkGoldenRod2',font='STIX')

userlvl = Label(win, text = "Username :",bg='DarkGoldenRod2')

passwdlvl = Label(win, text = "Password  :",bg='DarkGoldenRod2')
```

```
user1 = Entry(win, textvariable = StringVar())

passwd1 = Entry(win, textvariable = IntVar().set(""))

enter = Button(win, text = "Enter", command = lambda: login(), bd = 0)

enter.configure(bg = "blue")

Title.place(x = 830, y = 80)

user1.place(x = 800, y = 160)

passwd1.place(x = 800, y = 240)

userlvl.place(x = 720, y = 160)

passwdlvl.place(x = 720, y = 240)

enter.place(x = 845, y = 350)

Log_img = Image.open("Login_image.jpg")
resize_image = Log_img.resize((700, 496))
img = ImageTk.PhotoImage(resize_image)
Log_img = Label(win,image=img)
Log_img.place(relx=0.0,rely=0.0)

win.mainloop()
```

B_login(Buyer Login/Signup)

```
import mysql.connector as sql

from tkinter import *

from PIL import Image,ImageTk

import os

import sys


if sys.argv:

    filepath = sys.argv[0]

    folder, filename = os.path.split(filepath)

    os.chdir(folder)



win = Tk()


win.configure(bg='DarkGoldenRod2')


win.geometry("1000x500")


win.title("Login Page")



def login() :


    db = sql.connect(host = "localhost", user = "root", passwd = "778240")
    cur = db.cursor()



    try :

        cur.execute("create database login")

        db = sql.connect(host = "localhost", user = "root", passwd = "778240",
database = "login")

        cur = db.cursor()



    except sql.errors.DatabaseError:

        db = sql.connect(host = "localhost", user = "root", passwd = "778240",
database = "login")

        cur = db.cursor()



        try :

            cur.execute("create table main(username varchar(50), NOT NULL,
password int NOT NULL)")

        except sql.errors.ProgrammingError:
            pass



    finally :

        try :

            cur.execute("create table main(username varchar(50) NOT NULL, "

                        "password int NOT NULL)")



        except sql.errors.ProgrammingError:

            pass

    while True :

        user = user1.get()

        passwd = passwd1.get()

        cur.execute("select * from main where username = '%s' and password = %s"
% (user, passwd))
        rud = cur.fetchall()

        if rud:
            print("Welcome")

            break

        else:

            cur.execute("insert into main values('{}', {})".format(str(user), passwd))

            db.commit()

            print("Account Created")

            break

    cur.close()
    db.close()
    win.destroy()
    import buying

Title = Label(win, text = "Login",bg='DarkGoldenRod2',font='STIX')

userlvl = Label(win, text = "Username :",bg='DarkGoldenRod2')

passwdlvl = Label(win, text = "Password  :",bg='DarkGoldenRod2')
user1 = Entry(win, textvariable = StringVar())
```

```
passwd1 = Entry(win, textvariable = IntVar().set(""))

enter = Button(win, text = "Enter", command = lambda: login(), bd = 0)

enter.configure(bg = "blue")

Title.place(x = 830, y = 80)

user1.place(x = 800, y = 160)

passwd1.place(x = 800, y = 240)

userlvl.place(x = 720, y = 160)

passwdlvl.place(x = 720, y = 240)

enter.place(x = 845, y = 350)

user = user1.get()

Log_img = Image.open("Login_image.jpg")
resize_image = Log_img.resize((700, 496))
img = ImageTk.PhotoImage(resize_image)
Log_img = Label(win,image=img)
Log_img.place(relx=0.0,rely=0.0)

win.mainloop()
```

Selling (Image upload window)

```
from msilib.schema import File

import mysql.connector as sql

from tkinter import *

from PIL import Image,ImageTk

import os

import sys


from mysqlx import SqlStatement


if sys.argv:

    filepath = sys.argv[0]

    folder, filename = os.path.split(filepath)

    os.chdir(folder)


Sell = Tk()

Sell.configure(bg='Aquamarine')


Sell.geometry('1000x500')


image = Image.open("camera1.png")


resize_image = image.resize((200, 100))


img = ImageTk.PhotoImage(resize_image)
```

```
def listing():

    item = Tk()

    item.geometry('300x300')

    userlvl = Label(item, text = "Username :")
    userlvl.place(x = 50, y = 80)

    user1 = Entry(item, textvariable = StringVar())
    user1.place(x = 150, y = 80)

    Itm1 = Label(item, text = 'Enter file path:')
    Itm1.place(x = 50, y = 120)

    Item1 = Entry(item, textvariable = StringVar())
    Item1.place(x = 150, y = 120)

    def query1():

        def insert_blob(FilePath):
            with open(FilePath, "rb") as File:
                BinaryData = File.read()

            SQLStatement = "Insert into images(photo) values(%s)"
            cur.execute(SQLStatement, (BinaryData, ))
            db.commit()

        I1 = Item1.get()

        db = sql.connect(host = "localhost", user = "root", passwd =
"778240",database = "login")
        cur = db.cursor()

        insert_blob(I1)
        db.commit()

    enter = Button(item, text = "Enter", command = query1, bd = 0)
    enter.configure(bg = "blue")
    enter.place(x = 138, y = 175)

Listing = Button(Sell, image = img,command=listing, width= 200)
Listing.place(x=50,y=50)

def Retrieve_blob(ID):

        db = sql.connect(host = "localhost", user = "root", passwd =
        "778240",database = "login")
        cur = db.cursor()
        SQLStatement = "select * from images where ID ='{0}'"
        cur.execute(SQLStatement.format(str(ID)))
        result = cur.fetchone()[1]
        StoreFilePath = "Image_outputs./{0}.jpg".format(str(ID))
        print(result)
        with open(StoreFilePath, 'wb') as File:
            File.write(result)
            File.close()

'''db = sql.connect(host = "localhost", user = "root", passwd = "778240",database
= "login")
cur1 = db.cursor()
query = "Select max(id) from images"
cur1.execute(query)
myresult = cur1.fetchall()
Retrieve_blob(1)'''
Sell.mainloop()
```

Buying (Buyer window to purchase items)

```
from msilib.schema import File
import mysql.connector as sql
from tkinter import *
from PIL import Image,ImageTk
import os
import sys

from mysqlx import SqlStatement



if sys.argv:
    filepath = sys.argv[0]
    folder, filename = os.path.split(filepath)
    os.chdir(folder)

x = True
while x == True:

    Buy = Tk()
    Buy.configure(bg='Aquamarine')

    Buy.geometry('1050x500')

    def Page_destroy():
        global x
        x = False
        Buy.destroy()

    def BuyNow():

        win = Tk()

        win.configure(bg='goldenrod')

        win.geometry('500x500')
        l = Label(win,text='ITEM PURCHASED!',bg = 'goldenrod',font='STIX')
        l.place(relx=0.3,rely=0.4)
        def win_des():
            win.destroy()
        b = Button(win,text='Quit',command=win_des,width=20)
        b.place(relx=0.35,rely=0.8)


    Quit = Button(Buy, text= "Quit", command=Page_destroy, width= 20, bg =
'blue', fg = 'black')
    Quit.place(relx= 0.4 , rely= 0.9)

    def Mobile():
        Buy.destroy()

        Desc = Tk()
        Desc.configure(bg='pale green')
        Desc.geometry('500x500')

        def Desc_destroy():
            Desc.destroy()

        def End():
            global x
            x = False

        resize_image = mobile.resize((400, 200))
        img = ImageTk.PhotoImage(resize_image)

        label = Label(image=img)
        label.mobile = img
        label.place(x=50,y=50)

        back = Button(Desc, text= "back", command=Desc_destroy, width= 20, bg
        = 'blue', fg = 'black')
        back.place(relx= 0.0 , rely= 0.95)

        mb_desc = '\tTouchscreen Phone with \n \t  1) 108 megapixel camera \n2) 6
        Gb RAM \n      3) 64 Gb storage \n    Price: Rs 17500'
        desc = Label(Desc, text = mb_desc, bg = 'pale green', font='STIX')
        desc.place(relx=0.05,rely=0.6)

        buynow = Button(Desc,text = "BUY NOW",
command=lambda:[BuyNow(),Desc_destroy(),End()], width= 20)
        buynow.place(relx=0.7,rely=0.95)
```

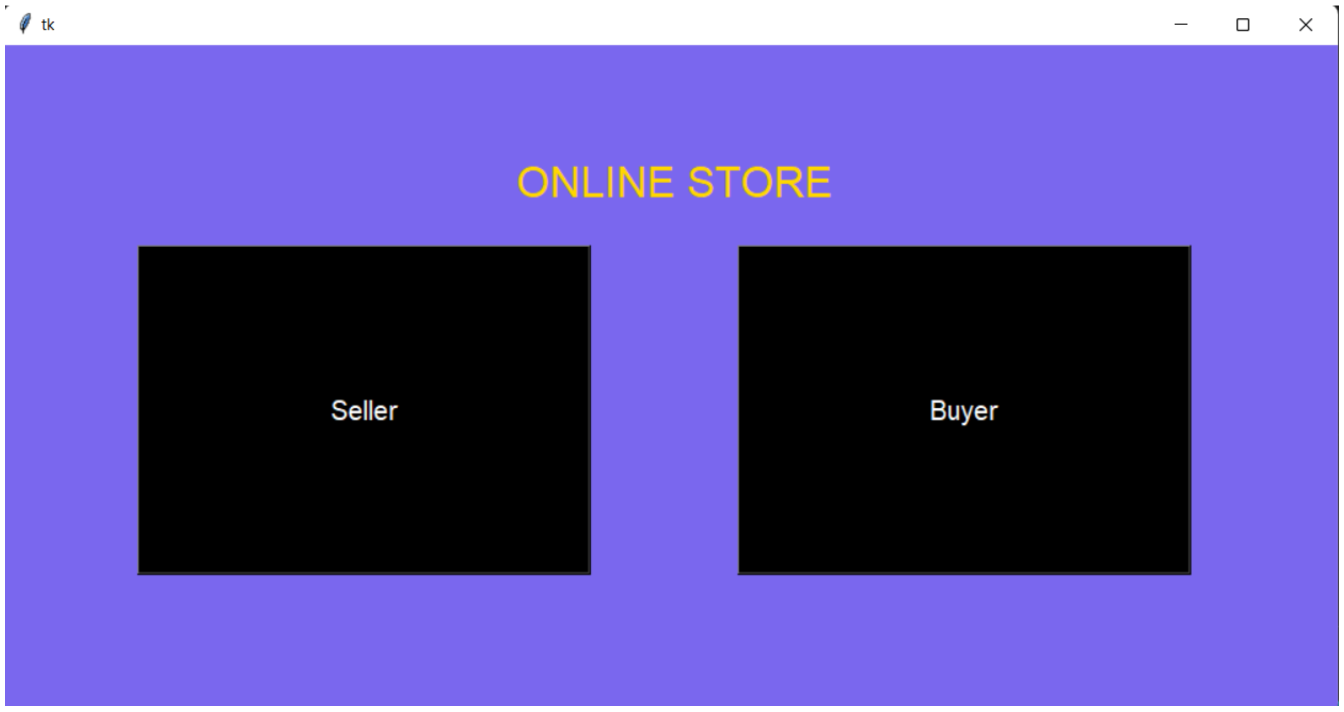
```
mobile = Image.open("mobile.jfif")
resize_image = mobile.resize((200, 100))
img = ImageTk.PhotoImage(resize_image)
Listing = Button(Buy, image = img, command=Mobile, width= 200)
Listing.place(x=50,y=50)

def Laptop():
    Buy.destroy()

    Desc = Tk()
    Desc.configure(bg='pale green')
    Desc.geometry('500x500')

    def Desc_destroy():
        Desc.destroy()

    def End():
        global x
        x = False

    resize_image = laptop.resize((400, 200))
    img1 = ImageTk.PhotoImage(resize_image)

    label1 = Label(image=img1)
    label1.laptop = img1
```

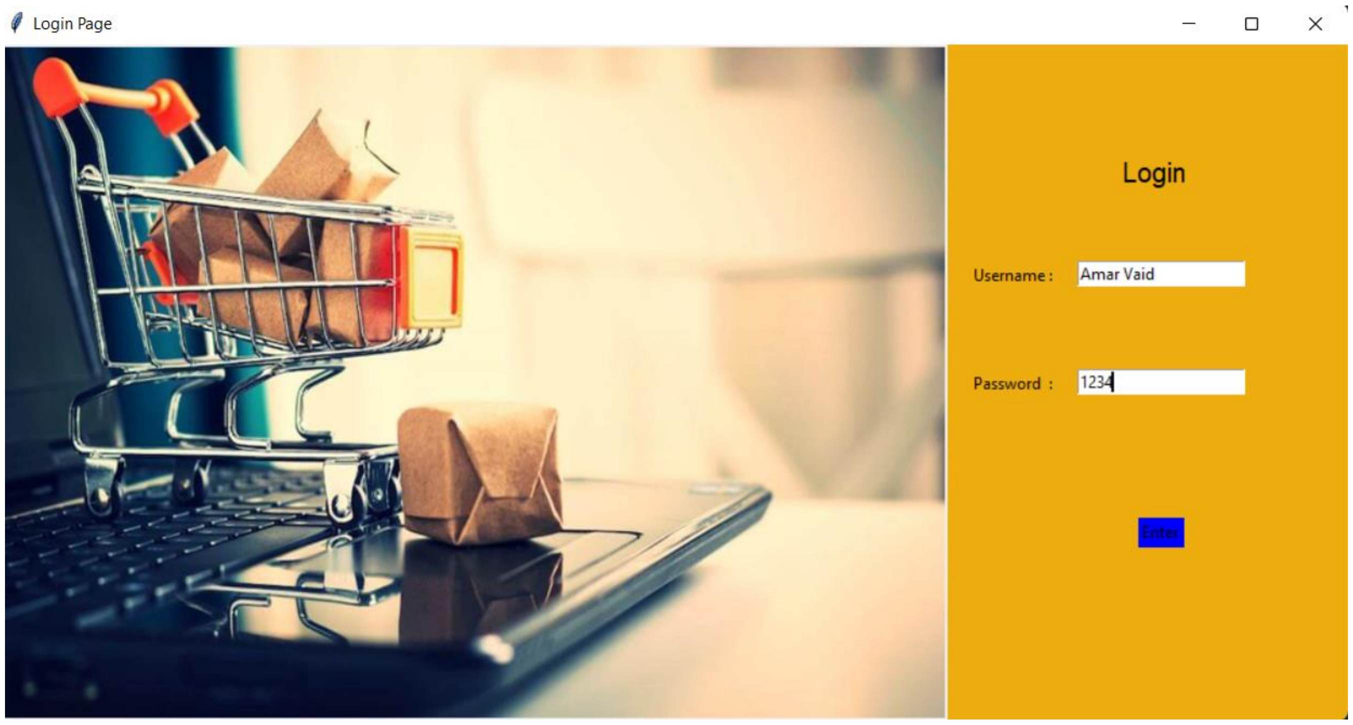
```
        label1.place(x=50,y=50)


        back = Button(Desc, text= "back", command=Desc_destroy, width= 20, bg
        = 'blue', fg = 'black')

        back.place(relx= 0.0 , rely= 0.95)


        mb_desc = '\t16 Inch Laptop with \n \t      1) 6 Gb Graphics Card  \n        2)
        16 Gb RAM \n        3) 1 Tb storage \n        Price: Rs 60000'

        desc = Label(Desc, text = mb_desc, bg = 'pale green', font='STIX')

        desc.place(relx=0.05,rely=0.6)


        buynow = Button(Desc,text = "BUY NOW",
command=lambda:[BuyNow(),Desc_destroy(),End()], width= 20)

        buynow.place(relx=0.7,rely=0.95)


        #enter description as label and add image with buy now option


    laptop = Image.open("laptop.jfif")

    resize_image = laptop.resize((200, 100))

    img1 = ImageTk.PhotoImage(resize_image)

    Listing1 = Button(Buy, image = img1, command=Laptop, width= 200)

    Listing1.place(x=300,y=50)


    def Camera():

        Buy.destroy()


        Desc = Tk()

        Desc.configure(bg='pale green')
Desc.geometry('500x500')

def Desc_destroy():
    Desc.destroy()

def End():
    global x
    x = False

resize_image = camera.resize((400, 200))
img2 = ImageTk.PhotoImage(resize_image)

label2 = Label(image=img2)
label2.camera = img2
label2.place(x=50,y=50)

back = Button(Desc, text= "back", command=Desc_destroy, width= 20, bg
= 'blue', fg = 'black')
back.place(relx= 0.0 , rely= 0.95)

mb_desc = '\tMirrorless Camera \n \t  1) Full Frame Sensor \n          2) 24
Megapixel \n         Price: Rs 35000'
desc = Label(Desc, text = mb_desc, bg = 'pale green', font='STIX')
desc.place(relx=0.05,rely=0.6)

buynow = Button(Desc,text = "BUY NOW",
command=lambda:[BuyNow(),Desc_destroy(),End()], width= 20)
buynow.place(relx=0.7,rely=0.95)
```

```
        #enter description as label and add image with buy now option


    camera = Image.open("camera01.webp")
    resize_image = camera.resize((200, 100))
    img2 = ImageTk.PhotoImage(resize_image)
    Listing2 = Button(Buy, text= "List Item", image = img2, command=Camera,
width= 200)
    Listing2.place(x=550,y=50)



    def Speakers():
        Buy.destroy()


        Desc = Tk()
        Desc.configure(bg='pale green')
        Desc.geometry('500x500')


        def Desc_destroy():
            Desc.destroy()


        def End():
            global x
            x = False


        resize_image = speakers.resize((400, 200))
        img3 = ImageTk.PhotoImage(resize_image)
        label3 = Label(image=img3)
        label3.speakers = img3
        label3.place(x=50,y=50)

        back = Button(Desc, text= "back", command=Desc_destroy, width= 20, bg
        = 'blue', fg = 'black')
        back.place(relx= 0.0 , rely= 0.95)

        mb_desc = '\tDolby Speakers with \n \t  1) 5.1 Surround Sound \n\t 2) upto
        105 db sound \n       Price: Rs 15000'
        desc = Label(Desc, text = mb_desc, bg = 'pale green', font='STIX')
        desc.place(relx=0.05,rely=0.6)

        buynow = Button(Desc,text = "BUY NOW",
        command=lambda:[BuyNow(),Desc_destroy(),End()], width= 20)
        buynow.place(relx=0.7,rely=0.95)

        #enter description as label and add image with buy now option

    speakers = Image.open("speakers.jpg")
    resize_image = speakers.resize((200, 100))
    img3 = ImageTk.PhotoImage(resize_image)
    Listing3 = Button(Buy, text= "List Item", image = img3, command=Speakers,
width= 200)
    Listing3.place(x=800,y=50)

    mainloop()
```

Output

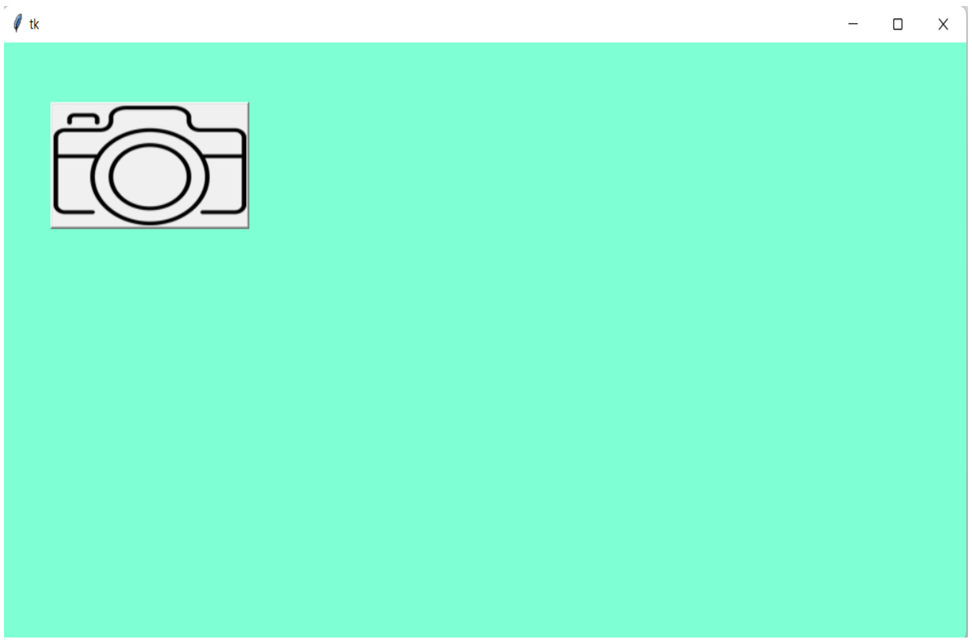
Seller

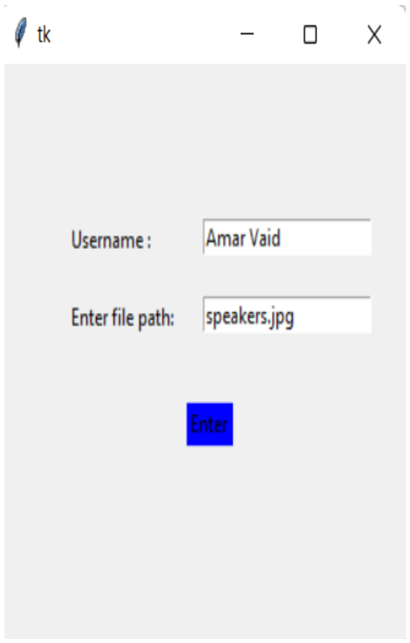




Username : Amar Vaid

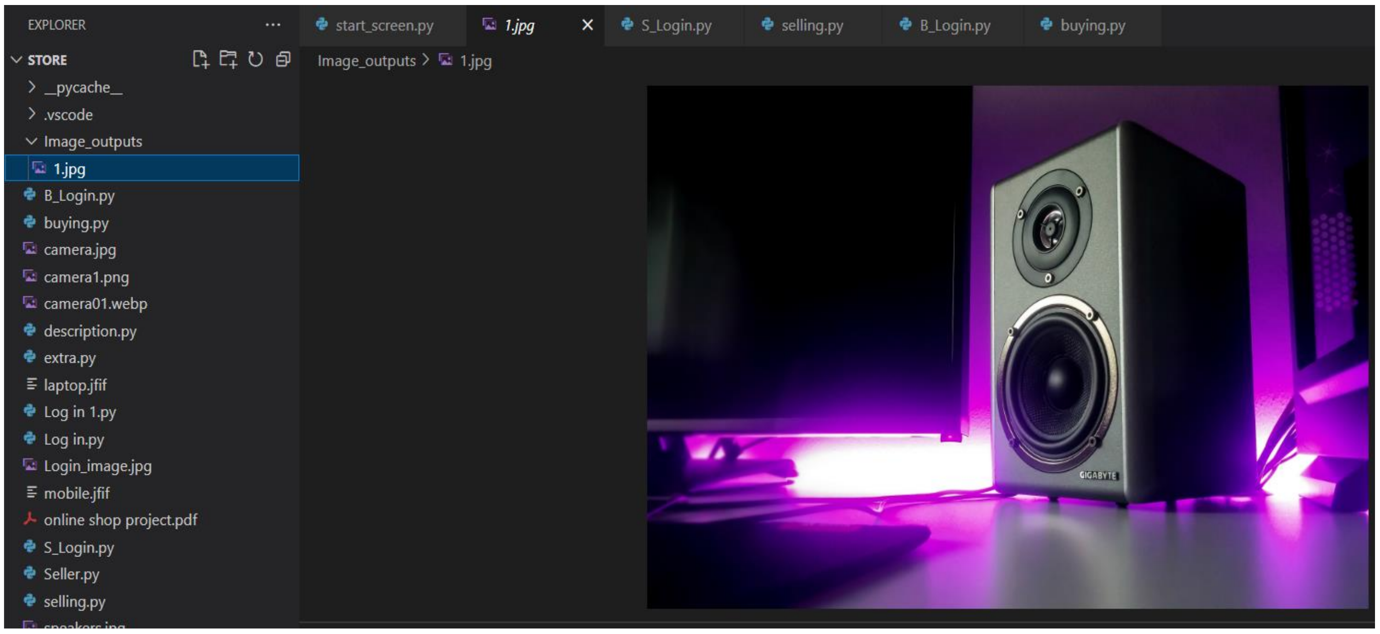
Password : 1234





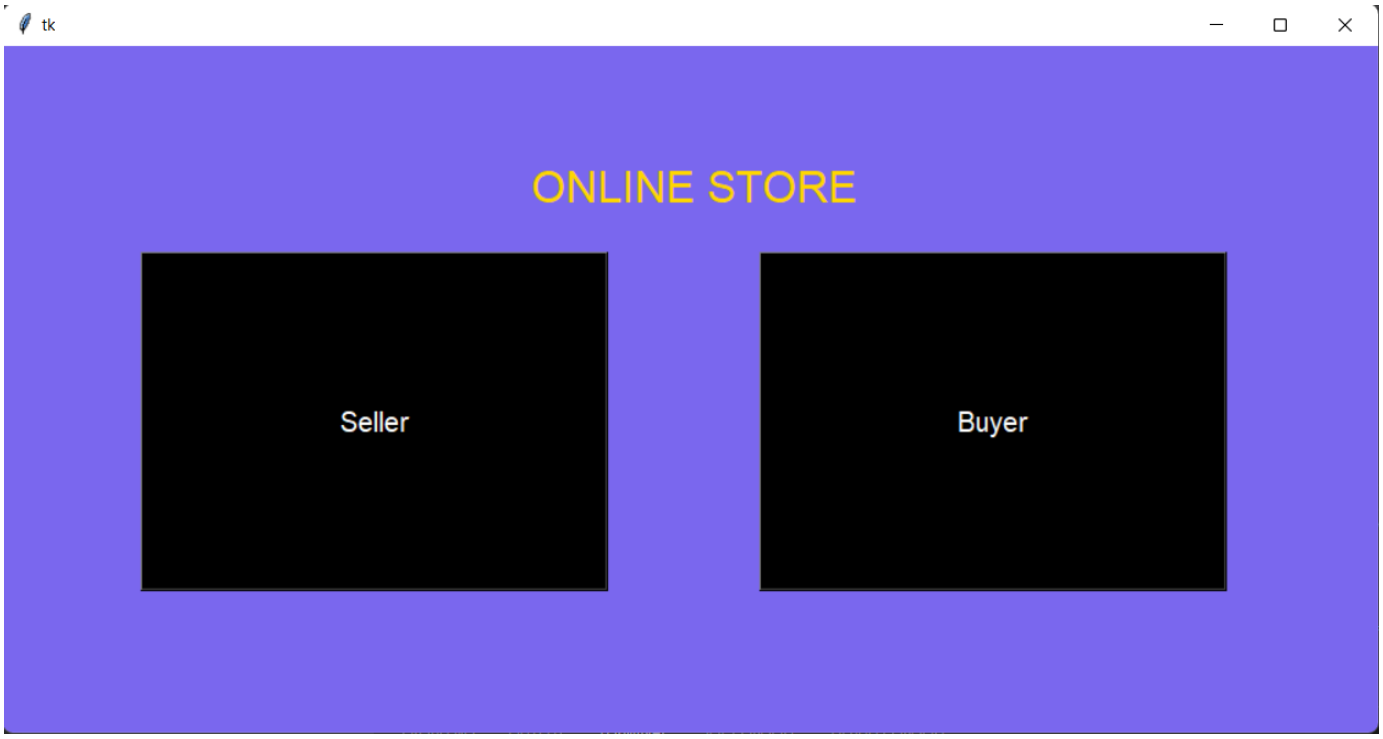
Username : Amar Vaid

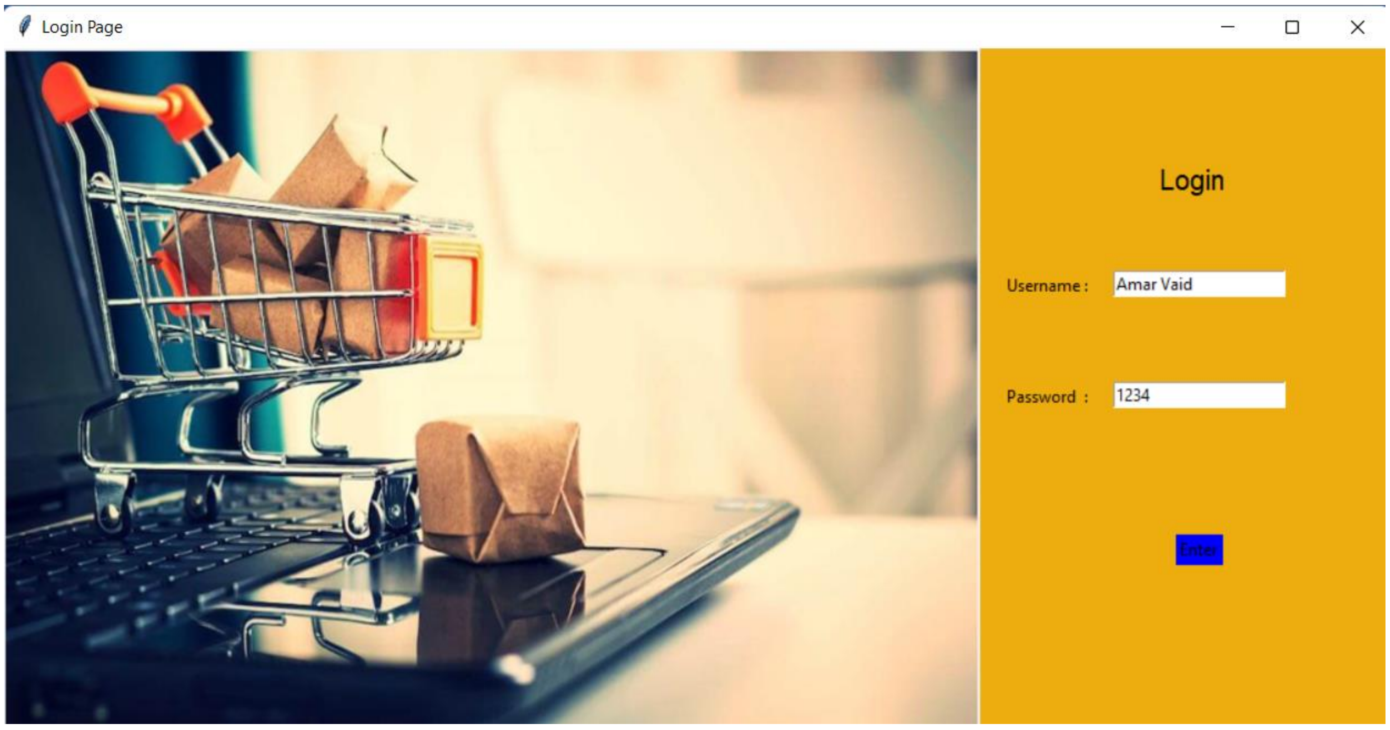
Enter file path: speakers.jpg



```
_pycache_
.vscode
Image_outputs
  1.jpg
B_Login.py
buying.py
camera.jpg
camera1.png
camera01.webp
description.py
extra.py
laptop.jfif
Log in 1.py
Log in.py
Login_image.jpg
mobile.jfif
online shop project.pdf
S_Login.py
Seller.py
selling.py
```

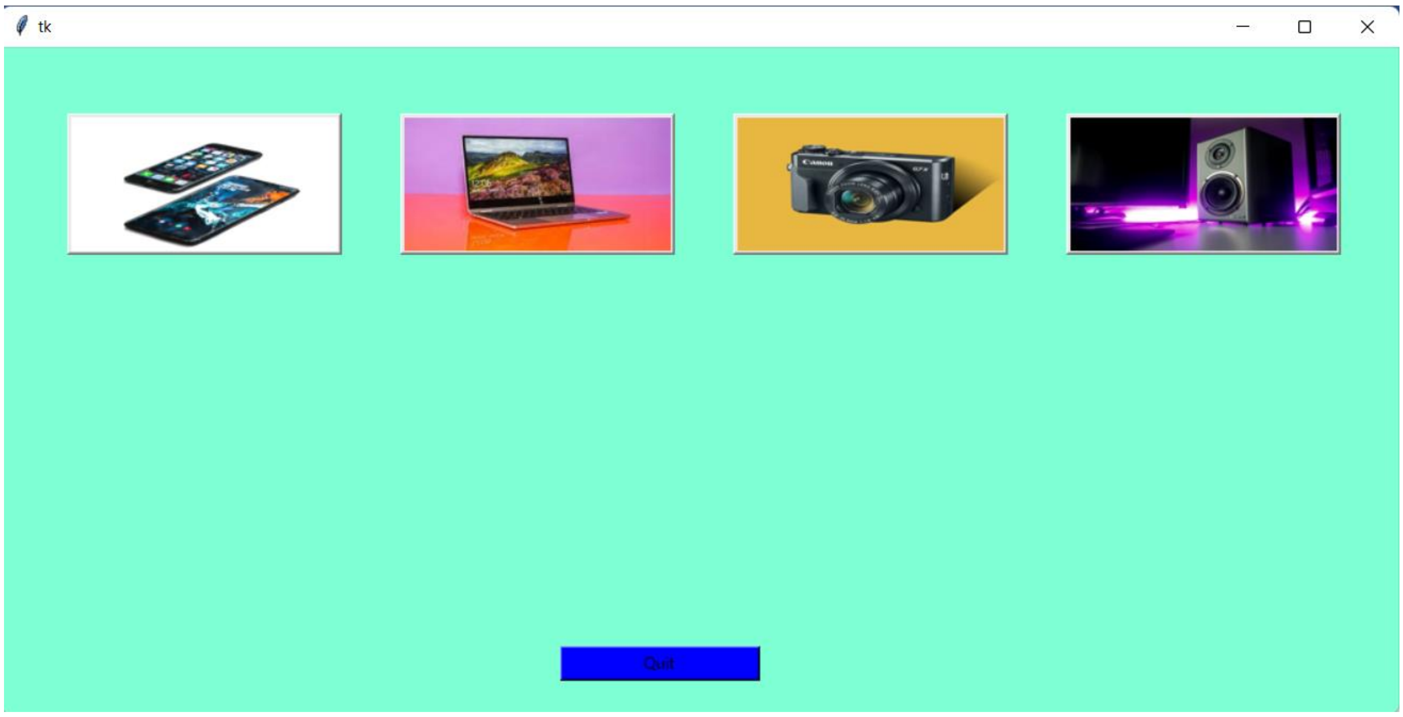
Buyer

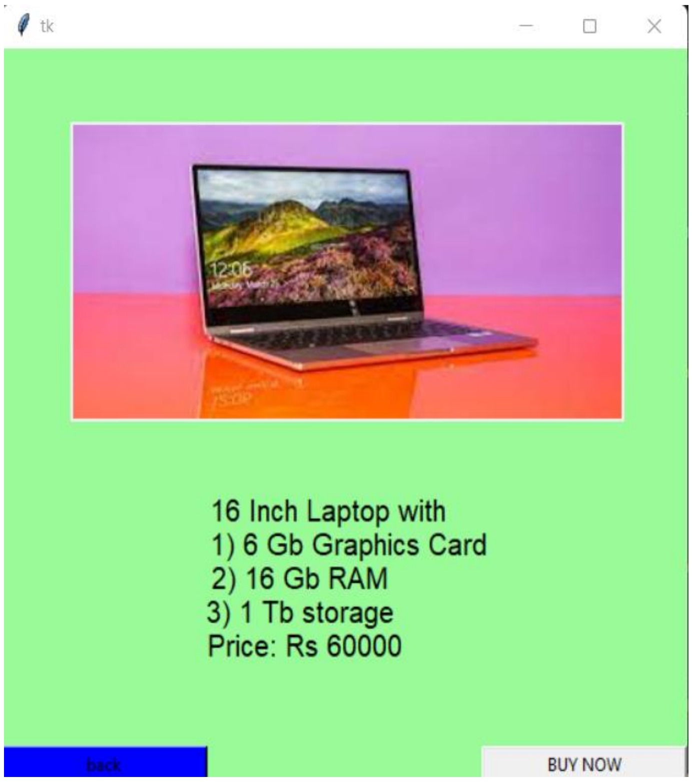




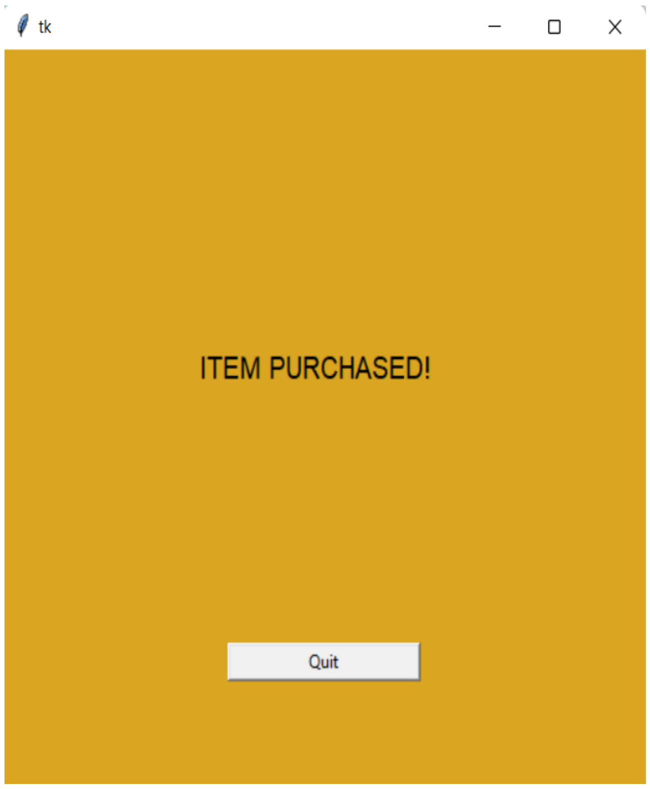
Username : Amar Vaid

Password : 1234





16 Inch Laptop with
1) 6 Gb Graphics Card
2) 16 Gb RAM
3) 1 Tb storage
Price: Rs 60000



ITEM PURCHASED!

Conclusion

There is a great scope in the future for this project as it has helped customers order various things they would need from the comfort of their houses. The use of python modules such as Tkinter and MySQL makes the program more attractive and easy to store data. This was also a learning experience on how to make software for bigger online store companies and how to store and manage data. This project can be further improved and made better for a larger customer range and more products.

Bibliography

Textbooks

- Computer Science with Python – Textbook for Class 12 – Sumita Arora

Websites Referred:

- www.python.org
- https://amazon.in/
- https://flipcart.in/
- www.geeksforgeeks.org
- https://stackoverflow.com/