CONSUMER PROFITABILITY ADVANCEMENTS BY PRICE ANALYSIS AND WEB SCRAPING IN E-COMMERCE

A PROJECT REPORT

Submitted by

Adeeba Rashid(20BAI10107)
Ipsita Maity(20BAI10159)
Nishita Gunjal(20BAI10186)
Nehal Jain(20BAI10242)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"CONSUMER PROFITABILITY ADVANCEMENTS BY PRICE ANALYSIS AND WEB SCRAPING IN E-COMMERCE"** is the bonafide work of **"Adeeba Rashid(20BAI10107), Ipsita Maity(20BAI10159) Nishita Gunjal(20BAI10186), Nehal Jain(20BAI10242)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. S. Sountharrajan, Associate Professor

School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. G. Elangovan, Assistant Professor

School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on _____

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express my heartfelt gratitude to **Dr. S. Sountharrajan** , Program Chair, School of Computer Science and Engineering for much of his valuable support encouragement in carrying out this work.

We would like to thank our internal guide **Dr. G. Elangovan** ,for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science and Engineering , who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

1. GUI – Graphical User Interface
2. RAM – Random Access Memory
3. Tkinter – TK Interface
4. E-commerce – Electronic Commerce

LIST OF FIGURES

ABSTRACT

- In this project, comparison of the price of a product from multiple eCommerce sites using Python will take place .
- The main objective is to monitor the price of the specific item (or items) from multiple eCommerce platforms.
- On each execution, all three websites are crawled and the product is located, and the price is obtained.

TABLE OF CONTENTS

References	

CHAPTER-1

PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

In this project, comparison of the price of a product from multiple eCommerce sites using Python will take place . The proposed project will be an application project which will compare the prices of a variety of electronics from a variety of brands.

1.2 Motivation for the work

- It is often seen in advertisements that there are websites which compares certain things like flight tickets (Make my Trip), policies (Policy Bazaar) and hotel rooms(Trivago) etc.
- However , one of the most ordered products fall in the electronics category .
- So , the proposed project will be an application project which will compare the prices of a variety of electronics from a variety of brands.

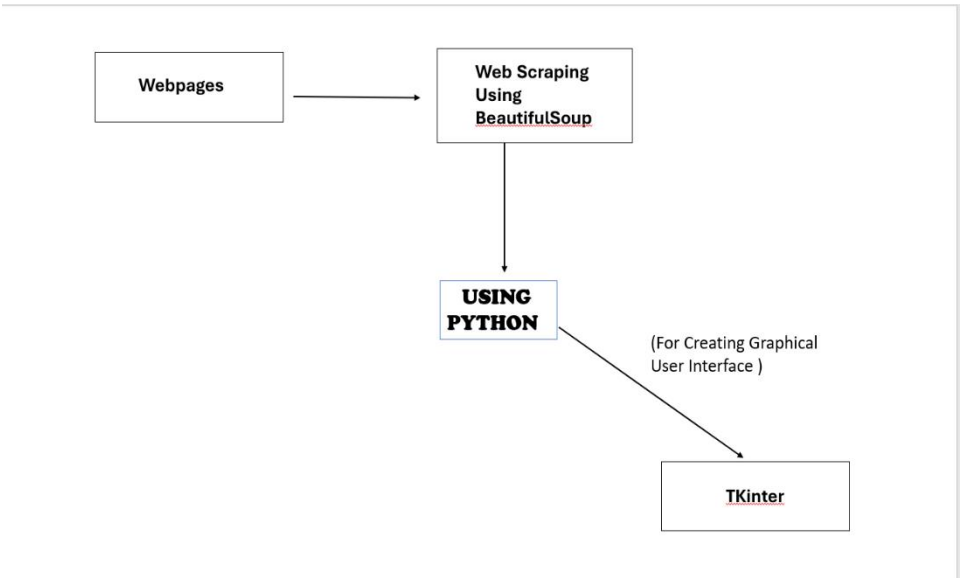
1.3 Problem Statement

- The purpose of price comparison project is to monitor the competitions prices, within their online store, in order to define a pricing strategy which is adapted to market changes.
- With the help of such a project, an online seller may:
- Know what prices are applied by their competitors to the products they also sell or similar.
- Know when competitor price changes take place.
- Obtain information about the distribution costs that can be covered by the prices your competitors charge.
- Detect opportunities to promote products with high demand.
- Control your product availability and compare it with your competitors in order to make correct decisions regarding stocking.

1.4 Objective of the work

- A web resource is intended to provide the user with key product information: its price, a brief description, availability in stores and more. Therefore, in addition to comparing prices, the consumer can also compare other important characteristics of the desired product/service.
- As statistics assure us, more than 80% of users take advantage of special websites to find the cheapest prices. Such web-based aggregators are useful to all participants in the process of buying a product or ordering a service.

1.5 Organization of the project



1.6 Summary

In this chapter an introduction to the proposed project was presented . This chapter also explained the motivation for work , the problem statement , objective of the work and the organization of the project . The proposed project is a price comparison project which will compare prices of different electronic goods from different e-commerce website using webscraping method .

CHAPTER – 2

RELATED WORK INVESTIGATION

2.1 Introduction

This chapter will focus on the investigation done for the related work . In this chapter the core area of the project will be discussed .

2.2 Core Area of the project

The proposed project has 3 core areas –

a) Python Programming Language - Python is an interpreted high-level general-purpose programming language.

b) Webscraping - Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites **.** BeautifulSoup Python library will be used for webscraping .

c) Graphical User Interface (GUI) - The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

2.3 Summary

Python programming language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. So, python is used in this project . Python provides both the features webscraping and GUI .

CHAPTER-3

REQUIREMENT ARTIFACT

3.1 INTRODUCTION

For our project to entirely come together, we have used a variety of applications and concepts like Tkinter and web scraping. We have mainly used python programming language and it's libraries to collect, operate and produce data in an organized manner and achieve the desired results of our project.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

1. **Laptop :** A requirement of a laptop with at least 8GB of RAM .

3.2.2 SOFTWARE REUIREMENTS

1. Python 3.7 version: Python programming language consists of a wide variety easy accessible libraries to use.

3.3 SPECIFIC PROJECT REQUIREMENTS

3.3.1 DATA REQUIREMENTS :

- For data requirements we have used a library in python known as Beautiful Soup.

- Beautiful Soup helps us to perform a concept called Web Scraping. It is data scraping used for extracting data from websites.

- In this project web scraping is being used to collect prices from a variety of e-commerce websites.

3.3.2 LOOK AND FEEL REQUIREMENTS

- To show our output in an organized matter we have used Tkinter library in python.

- The Tkinter library creates a GUI interface, in our project we use it show the product name and compare their prices (collected from e-commerce websites via web scraping.)

3.4 SUMMARY: This chapter covers all the basic requirements

covering all the spans (like data and look and feel) of the project .

CHAPTER 4

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and Goal:

Our project is following this methodology:

- Firstly we will take the prices of electronic gadgets from a variety of e-commerce websites.
- Secondly the collected data is used for comparison and is displayed on GUI interface using Tkinter.
- The goal is to provide customers a proper comaprison so that they can make the right choice while buying any electronic goods.

4.2 Software Architectural designs

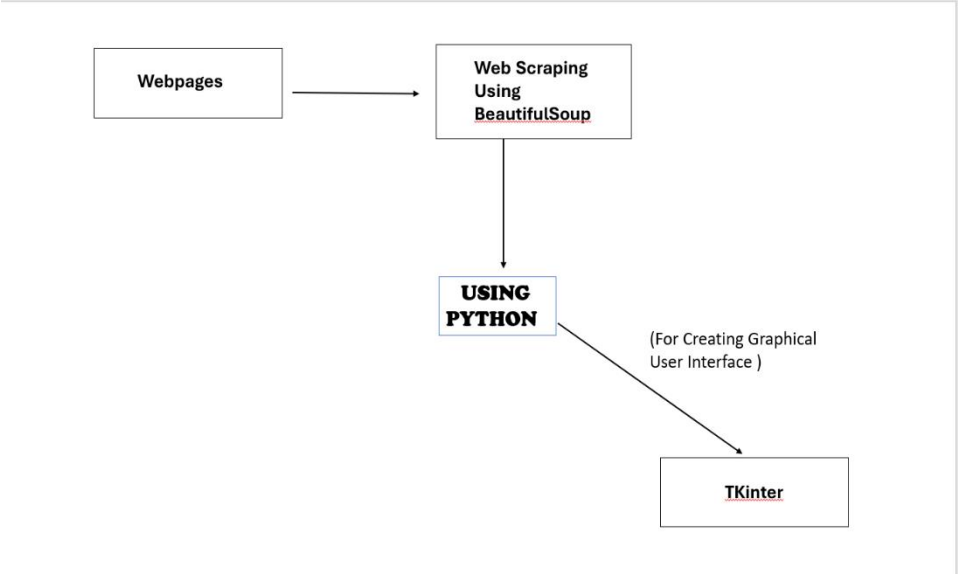


Figure 1

4.3 User Interface design

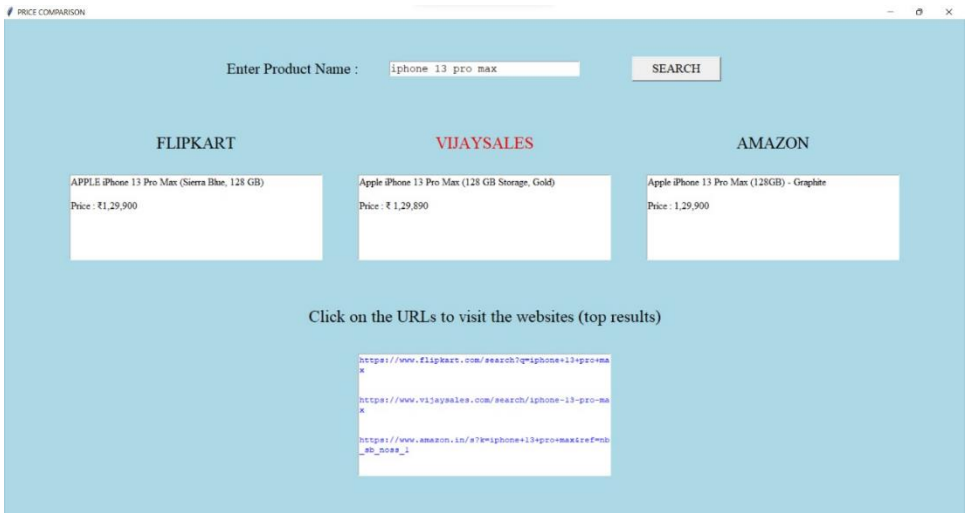


Figure 2

4.4 SUMMARY: This chapter covers the basic goals and basic planning of the proposed project. It also covers the basic method to do the project.

CHAPTER 5

TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Technical coding and code solutions

price comparison_web scraping.py > compare

```python
import requests
from bs4 import BeautifulSoup
from tkinter import *
from tkinter import Scrollbar
import webbrowser

global user_search
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck

def amazon(user_search):

    try:
        global amazon_url
        global amazon_price
        user_search1 = user_search.replace(' ','+')
        amazon_url = f"https://www.amazon.in/s?k={user_search1}&ref=nb_sb_noss_1"

        r = requests.get(amazon_url, headers = headers)
        soup = BeautifulSoup(r.text, 'html.parser')

        amazon_list = soup.select('.a-color-base.a-text-normal')
        amazon_page_length = int(len(amazon_list))


        for i in range(0,amazon_page_length):

            user_search = user_search.lower()
            amazon_id = amazon_list[i].getText().strip().lower()

            if any(x in amazon_id for x in user_search.split()):
                amazon_id = amazon_list[i].getText().strip()
                amazon_price = soup.select('.a-price-whole')[i].getText().strip()
                break

        return f"{amazon_id}\n\nPrice : {amazon_price}"

    except:
        amazon_price = '-'
        return 'Product not found'
```

```
        amazon_page_length = int(len(amazon_list))


        for i in range(0,amazon_page_length):

            user_search = user_search.lower()
            amazon_id = amazon_list[i].getText().strip().lower()

            if any(x in amazon_id for x in user_search.split()):
                amazon_id = amazon_list[i].getText().strip()
                amazon_price = soup.select('.a-price-whole')[i].getText().strip()
                break

        return f"{amazon_id}\n\nPrice : {amazon_price}"

    except:
        amazon_price = '-'
        return 'Product not found'
```

```
def flipkart(user_search):

    try:
        global flipkart_url
        global flipkart_price
        user_search1 = user_search.replace(' ','+')
        flipkart_url = f'https://www.flipkart.com/search?q={user_search1}'

        r = requests.get(flipkart_url, headers = headers)
        soup = BeautifulSoup(r.text, 'html.parser')

        flipkart_list = soup.select('._4rR01T')
        flipkart_page_length = int(len(flipkart_list))

        for i in range(0,flipkart_page_length):

            user_search = user_search.lower()
            flipkart_id = flipkart_list[i].getText().strip().lower()

            if any(x in flipkart_id for x in user_search.split()):
                flipkart_id = flipkart_list[i].getText().strip()
                flipkart_price = soup.select('._1_WHN1')[i].getText().strip()
```

```python
def vijaysales(user_search):

    try:
        global vijaysales_url
        global vijaysales_price
        user_search1 = user_search.replace(' ','-')
        vijaysales_url = f'https://www.vijaysales.com/search/{user_search1}'

        r = requests.get(vijaysales_url, headers = headers)
        soup = BeautifulSoup(r.text, 'html.parser')

        vijaysales_list = soup.select('.Dynamic-Bucket-ProductName')
        vijaysales_page_length = int(len(vijaysales_list))

        for i in range(0,vijaysales_page_length):

            user_search = user_search.lower()
            vijaysales_id = vijaysales_list[i].getText().strip().lower()

            if any(x in vijaysales_id for x in user_search.split()):
```

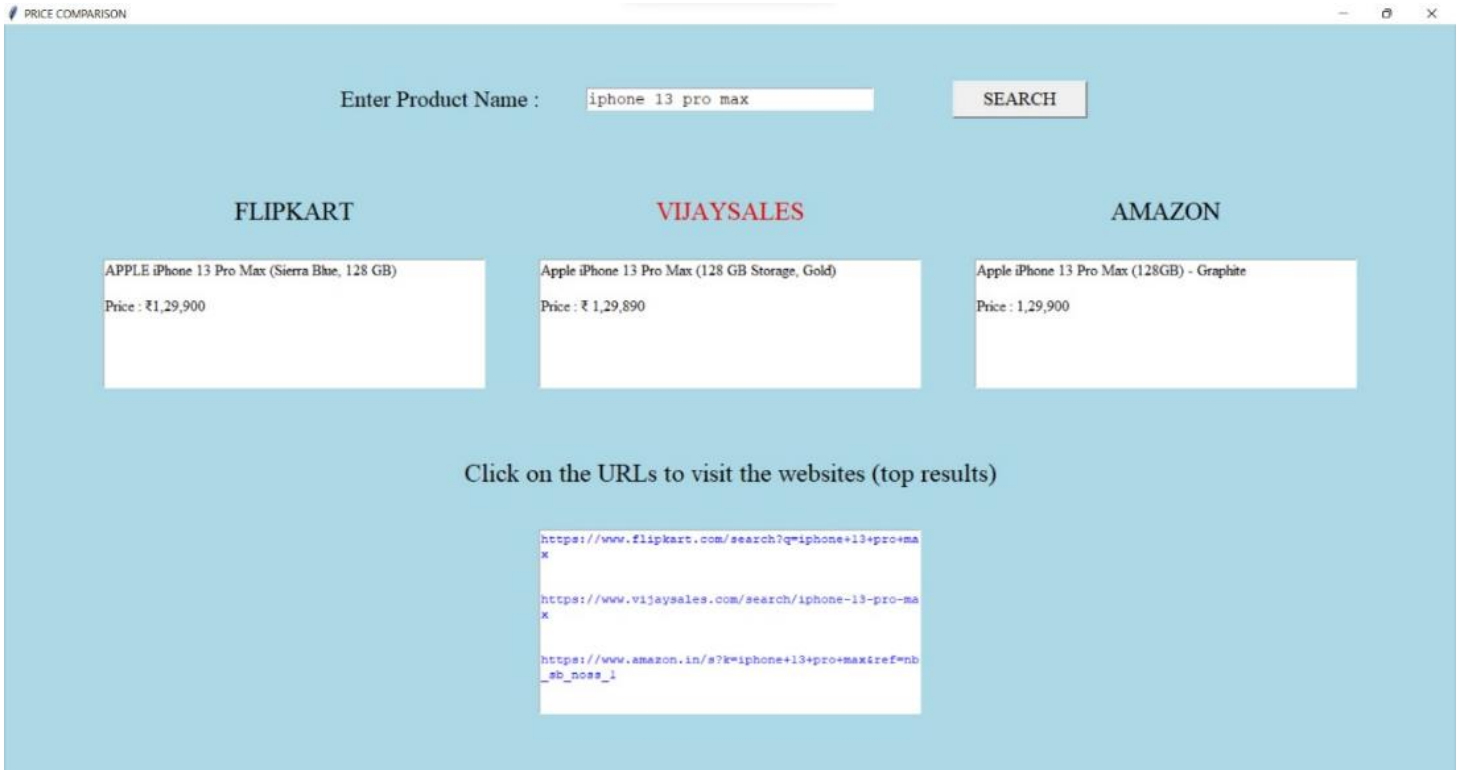
```python
        r = requests.get(vijaysales_url, headers = headers)
        soup = BeautifulSoup(r.text, 'html.parser')

        vijaysales_list = soup.select('.Dynamic-Bucket-ProductName')
        vijaysales_page_length = int(len(vijaysales_list))

        for i in range(0,vijaysales_page_length):

            user_search = user_search.lower()
            vijaysales_id = vijaysales_list[i].getText().strip().lower()

            if any(x in vijaysales_id for x in user_search.split()):
                vijaysales_id = vijaysales_list[i].getText().strip()
                vijaysales_price = soup.select('.Dynamic-Bucket-vsp')[i].getText().strip()
                break

        return f"{vijaysales_id}\n\nPrice : {vijaysales_price}"

    except:
        vijaysales_price = '-'
        return 'Product not found'
```

```
def compare():

    global l
    l = []

    ap = convert(amazon_price)
    fp = convert(flipkart_price)
    vsp = convert(vijaysales_price)

    # print(l)
    try:
        min_price = min(l)
        if min_price == fp:
            l1.config(fg = 'red2')
        elif min_price == vsp:
            l2.config(fg = 'red2')
        elif min_price == ap:
            l3.config(fg = 'red2')
        else:
            pass
    except:                                 # when the list is empty
        pass
```

```
def search():
    box1.insert(1.0,"LOADING...")
    box2.insert(1.0,"LOADING...")
    box3.insert(1.0,"LOADING...")
    box4.insert(1.0,"LOADING...")

    # search_button.place_forget()

    box1.delete(1.0,"end")
    box2.delete(1.0,"end")
    box3.delete(1.0,"end")
    box4.delete(1.0,"end")

    l1.config(fg = 'black')
    l2.config(fg = 'black')
    l3.config(fg = 'black')

    t1=flipkart(product_name.get())
    box1.insert(1.0,t1)

    t2=vijaysales(product_name.get())
    box2.insert(1.0,t2)
```

```
window = Tk()
window.wm_title("PRICE COMPARISON")
window.minsize(1500,700)
window.configure(bg = 'light blue')

lable_one =Label(window, text="Enter Product Name :", font=("Times New Roman", 18), bg = 'light blue')
lable_one.place(relx=0.3, rely=0.1, anchor="center")

product_name = StringVar()
product_name_entry = Entry(window, textvariable=product_name, width=30, font=("courier", 13))
product_name_entry.place(relx=0.5, rely=0.1, anchor="center")

search_button =  Button(window, text="SEARCH", width=12, command= search, font = ("Times New Roman", 15))
search_button.place(relx=0.7, rely=0.1, anchor="center")


l1 =  Label(window, text="FLIPKART", font=("Times New Roman", 20), bg = 'light blue')
l2 =  Label(window, text="VIJAYSALES", font=("Times New Roman", 20), bg = 'light blue')
l3 =  Label(window, text="AMAZON", font=("Times New Roman", 20), bg = 'light blue')
l4 =  Label(window, text="Click on the URLs to visit the websites (top results)", font=("Times New Roman
```

5.2 Prototype submission



5.3 Summary

So , in this chapter the implementation of technical coding is shown with the output .

CHAPTER 6

PROJECT OUTCOME AND APPLICABILITY

6.1 OUTLINE

- Before consumers decide in favor of a particular product or service, they compare offers from different providers and sellers to find the most profitable solution. It is best to do that using price comparison websites aggregating lots of apps in one place.
- Such projects bring high profits, and do not require large development or content creation costs. Users can always find an interesting and promising niche beyond the main trends and build their own price comparison website.
- Visual effect of product during manually purchasing the product is different from viewing the product in their device. This application requires active internet connection. User need to put correct data or else it behaves abnormally.

6.2 PROJECT OUTCOMES

- Significant outcome is that it helps users to find better deals on the same products or find similar products at better prices.

6.3 REAL WORLD APPLICATION

- This project is used in e-commerce websites like flipkart, amazon and chroma in real world.

6.4 INFERENCE

- Setting a price based on the perceived value of user's products and service in the minds of customers as opposed to factors such as costs. So, this project will save time of users as well as benefit them.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATION

7.1 OUTLINE

- This project basically compares two products in a way that the readers can grasp instantly. Having a Product Comparison enables user and visitors to see if featured items or products are better than competitor companies or businesses.

7.2 LIMITATIONS

- Visual effect of product during manually purchasing the product is different from viewing the product in your device.
- This application requires active internet connection.
- User need to put correct data or else it behaves abnormally.

7.3 FUTURE ENHANCEMENT

- This project is going to be big. With the growing number of marketplaces, that would help price comparison site to expand its service due to its relation. For those who are looking into buying products/brands in Marketplaces.

7.4 INFERENCE

- Consumers increasingly rely on Internet price comparison sites (PCS) to gain knowledge about the market. The prices generated by a PCS search can act as contextual reference prices and influence the attractiveness of prices encountered later as consumers shop offline at local stores.PCS retailer ratings is stronger for high-priced retailers and for consumers who rely less on the retailer price as a heuristic to infer retailer service level. So this project finds better deals on the same products or find similar products at better prices.

REFERENCES

1. Ryan Mitchell , Web scraping with python , *Collecting more data from the modern web* , 2018 , 978-1-491-98557-1

2. Alan D. Moore ,Python GUI programming with Tkinter ,*Develop responsive and powerful GUI applications with Tkinter* , 2018 , 971-1-78883-588-6

3. Wesley J. Chun, *Core Python Programming* ,2001 , 0-13-026036-3