SUNBEAM ENGLISH SCHOOL
BHAGWANPUR

SESSION: 2019-20

COMPUTER SCIENCE (085)

TOPIC: 'ITEM MANAGEMENT'

Name: ISHAN KUMAR

Class: 12-K

Board Roll No.: 23727294

CERTIFICATE

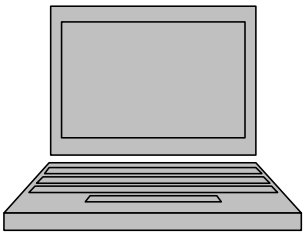
To Whom So Ever It May Concern

It is to certify that this project is the bona fide work of ISHAN KUMAR, class 12-K of Sunbeam English School, Bhagwanpur, Lanka, Varanasi. He has developed this Python project under my supervision, following strictly the guidelines of CBSE through the academic session 2019-2020.

Mr. Pradumna Singh

(PGT-Computer Sc.)

INDEX



TOPIC PAGE.NO

1.ACKNOWLEDGEMENT·····4

2.DIGITAL INDIA PROGRAMME·····5

3. HISTORY OF PYTHON·····14

2.PREFACE·····18

3.MODULES USED····· 19

5.USER DEFINED FUNCTION·····.19

6.PYTHON CODING·····21

7.SAMPLE OUTPUT·····27

8.LIMITATION·····31

9.BIBLIOGRAPHY·····32

ACKNOWLEDGEMENT

It gives me immense pleasure to present before you "***PRODUCT MANAGEMENT***" strictly in accordance to the guidelines given by C.B.S.E. The overwhelming response of our friends and the keen interest shown by my caring and loving teacher **Mr. Pradumna Singh,** who promoted me to honour my commitment for making the project up to date.

I have made sincere effort to make the project more meaningful, complete, compact and comprehensive. It is a great pleasure to let you know that I have put my feelings into practice.

I also express my sincere thanks to other experience & respected computer faculties ***Vinay Sir, Dhananjay Sir, Krishna Bhagwan Mishra Sir & Kaushal Srivastava Sir*** for their blessings & valuable suggestions which have made my project complete. I also want to thank hardware engineers ***Mr. Nitin Saxena, Mr. Hemant & Mr. Keval, Lab Attendant Mr. Abhijeet Dey & Lab Boy Mr. Ashish*** for their technical & moral support.

Finally, gracious thanks to my family members & friends, especially my parents, who have been perennial sources of inspiration.

ISHAN KUMAR

DIGITAL INDIA PROGRAMME

Digital India is an initiative of the Government of India to integrate the government departments and the people of India. It aims at ensuring that the government services are made available to citizens electronically by reducing paperwork. The initiative also includes a plan to connect rural areas with high-speed internet networks.

The creation of digital infrastructure

1. A programme to transform India into digital empowered society and knowledge economy.
2. The vision of Digital India aims to transform the country into a digitally empowered society and knowledge economy.
3. The programme will be implemented in phases from the current year till 2020

A **two-way platform** will be created where both the service providers and the consumers stand to benefit. The scheme will be monitored and controlled by the **Digital India Advisory group** which will be chaired by the Ministry of Communications and IT. It will be an **inter-ministerial initiative** where all ministries and shall offer their own services to the public Healthcare, Education, Judicial services etc. The Public-Private-Partnership model shall be adopted selectively. In addition, there are plans to restructure the **National Informatics Centre.** This project is one among the top priority projects of the Modi Administration.

Several events were held across 36 states and union territories, covering 600 districts in the country.

Information Technology companies were told to organise a mandatory viewing of the speech to be delivered by PM Narendra Modi at the launch of the Digital India Initiative.

What is Digital India?

With the launch of the Digital India programme, the government is taking a big step forward to transform the country into a digitally
empowered knowledge economy.

Scope of Digital India

1. To prepare India for a knowledge future.
2. On being transformative that is to realize IT (Indian Talent) + IT (Information Technology) = IT (India Tomorrow).
3. Making technology central to enabling change.

4. On being an Umbrella Programme –covering many departments.

The programme includes projects that aim to ensure that government services are available to citizens electronically and people get the benefit of the latest information and communication technology.

Apps for Digital India

Digital India Portal, MyGov Mobile App, Swachh Bharat Mission App and Aadhaar Mobile Update App

Vision of Digital India

Digital Infrastructure as a Utility to Every Citizen-Governance & Services on Demand Digital Empowerment of Citizens

Pillars Of Digital India

1. Broadband Highways
2. Universal Access to Phones
3. Public Internet Access Programme
4. e-Governance – Reforming government through Technology
5. e-Kranti – Electronic delivery of services
6. Information for All
7. Electronics Manufacturing – Target NET ZERO Imports
8. IT for Jobs
9. Early Harvest Programmes

Impact of Digital India by 2019

- Broadband in 2.5 lakh villages, universal phone connectivity
- Net Zero Imports by 2020
- 400,000 Public Internet Access Points
- Wi-Fi in 2.5 lakh schools, all universities; Public Wi-Fi hotspots for citizens

Digital Inclusion: 1.7 Cr trained for IT, Telecom and Electronics Jobs

Job creation: Direct 1.7 Cr. and Indirect at least 8.5 Cr

e-Governance & e-Services: Across government India to be a leader in IT use in services – health, education, banking

Digitally empowered citizens – public cloud, internet access

Benefits of Digital Locker

Digital Locker facility will help citizens to digitally store their **important documents** like PAN card, passport, mark sheets and degree certificates.

Digital Locker will provide **secure access to Government-issued documents**. It uses authenticity services provided by Aadhaar. It is aimed at eliminating the use of physical documents and enables sharing of verified electronic documents across Govt Agencies.

Digital Locker provides dedicated **personal storage space** in the cloud to citizens, linked to citizens Aadhaar number.

Digital Locker will **reduce the administrative overhead** of government departments and agencies created due to paperwork.

It will also make it easy for the residents to **receive services** by saving time and effort as their documents will now be available anytime, anywhere and can be shared electronically.

To **sign-up** for your Digital Locker, one need your Aadhaar number and a mobile number that is linked to that Aadhaar Number.

What is the National Optical Fibre Network (NOFN)?

NOFN proposes seven lakh kilometres of optical fibre to be laid to connect 250 gram panchayats in three years. **Public Wi-Fi spots** will be provided around the clusters after that and all villages will be provided with internet connectivity.

According to Communications and Information Technology Minister **Ravi Shankar Prasad**, "ten states including Maharashtra, Madhya Prasad, Rajasthan, West Bengal, Haryana and Chhattisgarh, are ready to roll out the NOFN to facilitate Digital India. States like Telangana, Meghalaya, Jharkhand have decided to observe Digital India Week (DIW) from July 1 to July 7.

The University Grants Commission (UGC) also directed all varsities and higher education institutes across the country to observe the Digital India Week.

Benefits of Digital India

1. It would bring in **public accountability** through mandated delivery of government's services electronically.

2. It will be done through a **Unique ID and e-Pramaan** based on authentic and standard-based interoperable and integrated government applications and data basis.

The Government of India has initiated a giant leap forward to transform the country into a digitally empowered knowledge economy. DI will help in leveraging India's Globally acclaimed **IT competence** for the benefit of 120 Crore Indians.

It will help in reducing corruption, getting things done quickly and will help in reducing paperwork.

Some of the **facilities** which would be available through this initiative are Digital Locker, e-education, e-health, Digital Signature and national scholarship portal.

Digital India: Challenges and opportunities

I would suggest the govt make 2015-16 the year of broadband highways and ensure at least 50,000 panchayats are truly connected and functional

We are in the middle of Digital India Week. Prime Minister Narendra Modi opened the week on 1 July with a kind of a function we have never seen before. The function at the Indira Gandhi indoor stadium was no less vibrant than corporate product launches. It had all the glitz of money, multimedia, videos, tall promises and, as a token, two women brought from villages to be handed over a laptop each in recognition of how they were using the Internet and computers to be successful entrepreneurs.

I also observed that the people invited to the event were the 1%—those who were connected, successful and do not need the help of any of those nine pillars on which Digital India is structured. All those living far away

in inaccessible areas in rural India who need to be connected were nowhere to be seen. The event merely mentioned that 40-odd panchayats were connected through video conferencing using the national optic fibre network (NOFN). Corporate announcements made at the event in the name of Digital India were not new investments, but existing plans which were now aligned to the programme. I would have appreciated if one of them had said that they would connect a certain percentage of India's 250,000 panchayats, or that they would provide digital literacy to a large number of villagers, or that they would adopt villages to make them broadband-connected

It is on record that not a single telecom operator or industry house has signed up to partner the NOFN programme, despite the department of telecom inviting them several times.

I have always liked the Digital India programme for one reason: It was hyped up so much that many businesses who had never thought of digital inclusion were now talking about it and, in many cases, deploying their charity spending on digital literacy or women's empowerment through digital tools or mobile phones.

Yet, when I sat through the entire course of the celebratory Digital India Week opening, two thoughts went through my mind.

One, why are we calling it a Digital India programme? It actually should have been called a Digital Bharat programme, considering

the challenges to connect the majority of the masses of India. We usually relate Bharat with our rural folk. Since the majority of the
population live in rural areas often called Bharat, we actually need a Digital Bharat programme to ensure that Bharat is as connected and digital as India, which lives in its metros and cities. Two, why are we calling it a Digital India Week?

Because to realize Digital India, and if we are serious about making the whole country digitally enabled, we need a Digital India decade, or even to create and keep the momentum sustainable and action-oriented, we need at least a Digital India Year.

A Digital India Year would make sure that each and every aspect of the programme is pushed to show results on the ground and not go into hibernation after one week of enthusiasm.

Considering that most of the nine pillars of the Digital India programme face serious challenges in implementation, it is imperative that focused, persistent attention be given to each
of its pillars so that the big programme does not end up in embarrassment and failure.

Considering that the Digital Empowerment Foundation has a footprint at more than 150 locations in India at the village level and that too with the purpose of digitally enabling the poorest of the poor, let me highlight some of the major challenges of the Digital India programme.

First and foremost is that the entire programme is designed as a top-down model. There is no idea of how it would be implemented on the ground to be successful. For example, let's consider six of the nine pillars of the programme directly related to consumers and people at large: broadband highways, universal access to mobile connectivity, public internet access programme, e-kranti or electronic delivery of services, information for all, and IT for jobs.

Broadband highways, now called BharatNet, is supposed to connect up to gram panchayat, but laying fibre optic cables is the least of the challenges here. The biggest challenge is ensuring that each panchayat point of broadband is fired up, functional, used and distributed. Our research work found that more than 67% of the NOFN points are nonfunctional, even at the pilot stage. Besides, if the broadband highways programme is not implemented well and soon, we may not be able to implement other pillars of the ability to implement other pillars of the programme, such as the public Internet access programme, e-kranti, information for all, and even IT for jobs. All these are dependent on access infrastructure.

Out of the many initiatives launched during the Digital India week, one that could make a serious impact is BSNL's (Bharat Sanchar Nigam Ltd) mass deployment of Wi-Fi hotspots across the country. If the government pushes BSNL to ensure at least one hotspot per panchayat or per village, it can do wonders, and the government can also show off this as a positive outcome. However, if the selection of the locations for the hotspots necessarily were
those populated by mostly tribals, backward castes, minorities and geographically difficult areas, then the impact would be something that would be a national story.

History of Python

Easy as ABC

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde & Informatica). The greatest achievement of ABC was to influence the design of Python.

Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum Voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."

Later on, in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties but without its

problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Comedy, Snake or Programming Language

So, what about the name "Python": Most people think about snakes, and even the logo depicts two snakes, but the origin of the name has its root in British humour. Guido van Rossum, the creator of Python, wrote in 1996 about the origin of the name of his programming language[1]: "Over six years ago, in December 1989, I was looking for a 'hobby' programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer and not much else on my hand. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus)."

The Zen of Python

- *Beautiful is better than ugly.*
- *Explicit is better than implicit.*
- *Simple is better than complex.*
- *Complex is better than complicated.*
- *Flat is better than nested.*
- *Sparse is better than dense.*
- *Readability counts.*
- *Special cases aren't special enough to break the rules.*

- *Although practicality beats purity.*
- *Errors should never pass silently.*
- *Unless explicitly silenced.*
- *In the face of ambiguity, refuse the temptation to guess.*
- *There should be one -- and preferably only one -- obvious way to do it.*
- *Although that way may not be obvious at first unless you're Dutch.*
- *Now is better than never.*
- *Although never is often better than *right* now.*
- *If the implementation is hard to explain, it's a bad idea.*
- *If the implementation is easy to explain, it may be a good idea.*
- *Namespaces are one honking great idea -- let's do more of those!*

Development Steps of Python

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object-oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.

Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode.

Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."

Some changes in Python 3.0:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

PREFACE

The modern age is the age of computers. The computer has become a part of life in the west where computer technology is making fast headway. It is essential to make evolutionary changes in every field, but in comparison among the developing countries, INDIA is developing very fast in the field of computer technology. The Prime Minister of India is also promoting Digital India motion.

The computer education in India started somewhere in the sixties at educational level. Now, with the invention of Supercomputers, it has become possible to have mass education in this field. The government of India has made an ambitious plan to make computer practically available to all the higher secondary schools in India at the end of the eighth year plan.

In India, the industries are now making the largest use of computers. State government is using computers for data processing, crime reporting & resource planning of the job.

Not only this, but it is also traditional data processing to be performed much faster. But within five years, computer technology has made tremendous progress in every field of work.

*I have developed this software namely "**ITEM MANAGEMENT**" using Python language. The creator of this software has left no stone unturned to convert it into an interactive & user-friendly way of managing products. It allows the user to enter new products, remove products, change product quantity/rating and easily display all the listed products.*

So let's start & see .....

Modules Used:

os

**For rename()*
**for remove()*

Functions by user:

- **main():** To initiate the program main menu.
- **addst():** Submenu to add new stationary. All details are requested from user and can be initiated by selecting option on menu.
- **remst():** Submenu to remove product. **Removal can only be done by item ID**. Can be initiated by selecting option on menu.
- **cquan():** Submenu to change quantity of item. **Changing quantity can only be done by item ID**. Can be initiated by selecting option on menu.

- **ccost():** Submenu to change cost of item. **Changing rating can only be done by item ID**. Can be initiated by selecting option on menu.
- **search():** Submenu to search for a item. **Searching can be done by item ID, name of item, item type, or by both name and type.** Can be initiated by selecting option on menu.
- **dispall():** Submenu to display all the items in an arranged way. Can be initiated by selecting option on menu.

PYTHON CODING

```
import pymysql as sq
import sys
def main():
   def addst():
      print("**********Add Stationary
Item**********")
      idp=int(input("Enter Item ID:"))
      n=input("Enter item Name:")
      t=input("Enter item type:")
      t=t.capitalize()
      q=int(input("Enter quantity of item:"))
      r=int(input("Enter cost"))
      qu="insert into item values({},'{}','{}',{},{})"
      myc.execute(qu.format(idp,n,t,q,r))
      print("Adding item id:",idp,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",r)
      b=input("Do you want to continue(Y/N):")
      if b in ['Y','y']:
         con.commit()
         print("Added item id:",idp,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",r)
         main()
      elif b in ['N','n']:
         con.rollback()
         print ("Process Cancelled\nExiting.....")
         main()
      else:
         print("Enter Valid Response")
         addst()

   def remst():
      print("**********Remove
item**********")
      a=int(input("Enter item Id:"))
      qu="delete from item where it_id={}"
      qu2="select * from item where it_id={}"
      myc.execute(qu2.format(a))
      cur=myc.fetchone()
      n,t,q,r=cur[1],cur[2],cur[3],cur[4]
      myc.execute(qu.format(a))
      print("Removing item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",r)
      b=input("Do you want to continue(Y/N):")
      if b in ['Y','y']:
         con.commit()
         print("Removed item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",r)
         main()
      elif b in ['N','n']:
```

```
            con.rollback()
            print ("Process Cancelled\nExiting.....")
            main()
        else:
            print("Enter Valid Response")
            remst()
    def cquan():
        print("**********************************Change item Quantity**********************")
        a=int(input("Enter item Id:"))
        qu="update item set quan={} where it_id={}"
        qu2="select * from item where it_id={}"
        myc.execute(qu2.format(a))
        cur=myc.fetchone()
        n,t,q,r=cur[1],cur[2],cur[3],cur[4]
        z=int(input("Enter New item Quantity:"))
        myc.execute(qu.format(z,a))
        print("Changing item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",q," To New
Quantity:",z,"\ncost:",r)
        b=input("Do you want to continue(Y/N):")
        if b in ['Y','y']:
            con.commit()
            print("Changed item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",z,"\ncost:",r)
            main()
        elif b in ['N','n']:
            con.rollback()
            print ("Process Cancelled\nExiting.....")
            main()
        else:
            print("Enter Valid Response")
            cquan()
    def ccost():
        print("**********************************Change item cost**************************")
        a=int(input("Enter item Id:"))
        qu="update item set cost={} where it_id={}"
        qu2="select * from item where it_id={}"
        myc.execute(qu2.format(a))
        cur=myc.fetchone()
        n,t,q,r=cur[1],cur[2],cur[3],cur[4]
        z=int(input("Enter New item cost:"))
        myc.execute(qu.format(z,a))
        print("Changing item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"cost:",r," to cost:",z)
        b=input("Do you want to continue(Y/N):")
        if b in ['Y','y']:
            con.commit()
            print("Changed item id:",a,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",z)
            main()
        elif b in ['N','n']:
            con.rollback()
            print ("Process Cancelled\nExiting.....")
            main()
        else:
            print("Enter Valid Response")
            crate()
    def search():
        print("**********************************Search item**************************")
```

```
    print("Search by==>\n1)item ID\n2)Name\n3)Type\n4)Advanced Search\n5)Exit To Menu")
    a=input("Enter choice to continue:")
    def idsearch():
        x=int(input("Enter item ID:"))
        qu2="select * from item where it_id={}"
        myc.execute(qu2.format(x))
        cur=myc.fetchone()
        n,t,q,r=cur[1],cur[2],cur[3],cur[4]
        print("item id:",x,"\nitem name:",n,"\nType:",t,"\nQuantity:",q,"\ncost:",r)
        main()
    def nsearch():
        x=(input("Enter Name to search:"))
        print("Sort By==>\n1)Quantity\n2)cost\n3)Do not sort")
        j=input("Enter choice to continue:")
        if j in ['1','1)']:
            print("1)Ascending\n2)Decending")
            k=input("Enter choice to continue:")
            if k in ['1','1)']:
                qu2="select * from item where name like '%{}%' order by quan asc"
                myc.execute(qu2.format(x))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif k in ['2','2)']:
                qu2="select * from item where name like '%{}%' order by quan desc"
                myc.execute(qu2.format(x))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        elif j in ['2','2)']:
            print("1)Ascending\n2)Decending")
            k=input("Enter choice to continue:")
            if k in ['1','1)']:
                qu2="select * from item where name like '%{}%' order by rate asc"
                myc.execute(qu2.format(x))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif k in ['2','2)']:
                qu2="select * from item where name like '%{}%' order by rate desc"
                myc.execute(qu2.format(x))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        elif j in ['3','3)']:
            qu2="select * from item where name like '%{}%'"
            myc.execute(qu2.format(x))
```

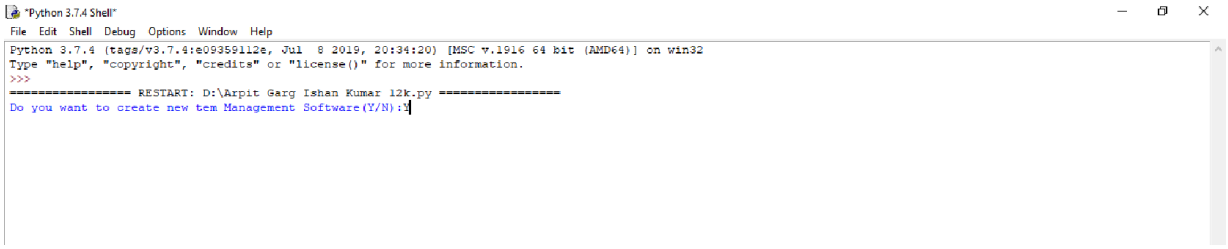
```
            rec=myc.fetchall()
            print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
            for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        def tsearch():
            x=(input("Enter Type to search:"))
            print("Sort By==>\n1)Quantity\n2)cost\n3)Do not sort")
            j=input("Enter choice to continue:")
            if j in ['1','1)']:
                print("1)Ascending\n2)Decending")
                k=input("Enter choice to continue:")
                if k in ['1','1)']:
                    qu2="select * from item where type like '%{}%' order by quan asc"
                    myc.execute(qu2.format(x))
                    rec=myc.fetchall()
                    print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                    for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
                elif k in ['2','2)']:
                    qu2="select * from item where type like '%{}%' order by quan desc"
                    myc.execute(qu2.format(x))
                    rec=myc.fetchall()
                    print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                    for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif j in ['2','2)']:
                print("1)Ascending\n2)Decending")
                k=input("Enter choice to continue:")
                if k in ['1','1)']:
                    qu2="select * from item where type like '%{}%' order by rate asc"
                    myc.execute(qu2.format(x))
                    rec=myc.fetchall()
                    print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                    for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
                elif k in ['2','2)']:
                    qu2="select * from item where type like '%{}%' order by rate desc"
                    myc.execute(qu2.format(x))
                    rec=myc.fetchall()
                    print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                    for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif j in ['3','3)']:
                qu2="select * from item where type like '%{}%'"
                myc.execute(qu2.format(x))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
```

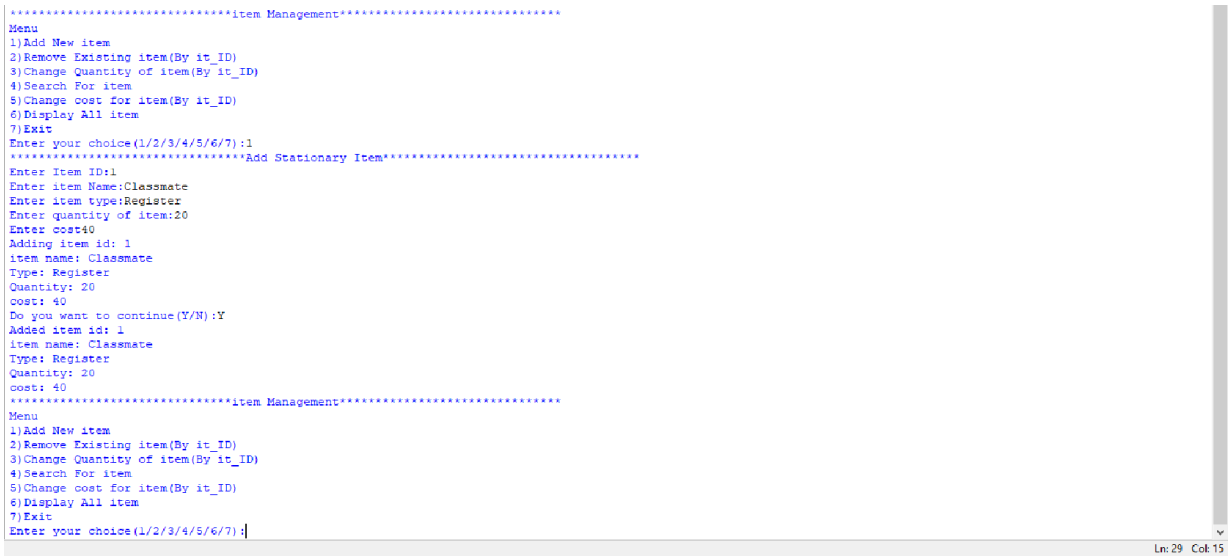
```
    def advsearch():
        x=(input("Enter Name to search:"))
        y=(input("Enter Type to search:"))
        print("Sort By==>\n1)Quantity\n2)cost\n3)Do not sort")
        j=input("Enter choice to continue:")
        if j in ['1','1)']:
            print("1)Ascending\n2)Decending")
            k=input("Enter choice to continue:")
            if k in ['1','1)']:
                qu2="select * from item where name like '%{}%' and type like '%{}%' order by quan asc"
                myc.execute(qu2.format(x,y))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif k in ['2','2)']:
                qu2="select * from item where name like '%{}%' and type like '%{}%' order by quan desc"
                myc.execute(qu2.format(x,y))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        elif j in ['2','2)']:
            print("1)Ascending\n2)Decending")
            k=input("Enter choice to continue:")
            if k in ['1','1)']:
                qu2="select * from item where name like '%{}%' and type like '%{}%' order by rate asc"
                myc.execute(qu2.format(x,y))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
            elif k in ['2','2)']:
                qu2="select * from item where name like '%{}%' and type like '%{}%' order by rate desc"
                myc.execute(qu2.format(x,y))
                rec=myc.fetchall()
                print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
                for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        elif j in ['3','3)']:
            qu2="select * from item where name like '%{}%' and type like '%{}%'"
            myc.execute(qu2.format(x,y))
            rec=myc.fetchall()
            print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
            for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
    if a in ['1','1)']:
        idsearch()
    elif a in ['2','2)']:
        nsearch()
```

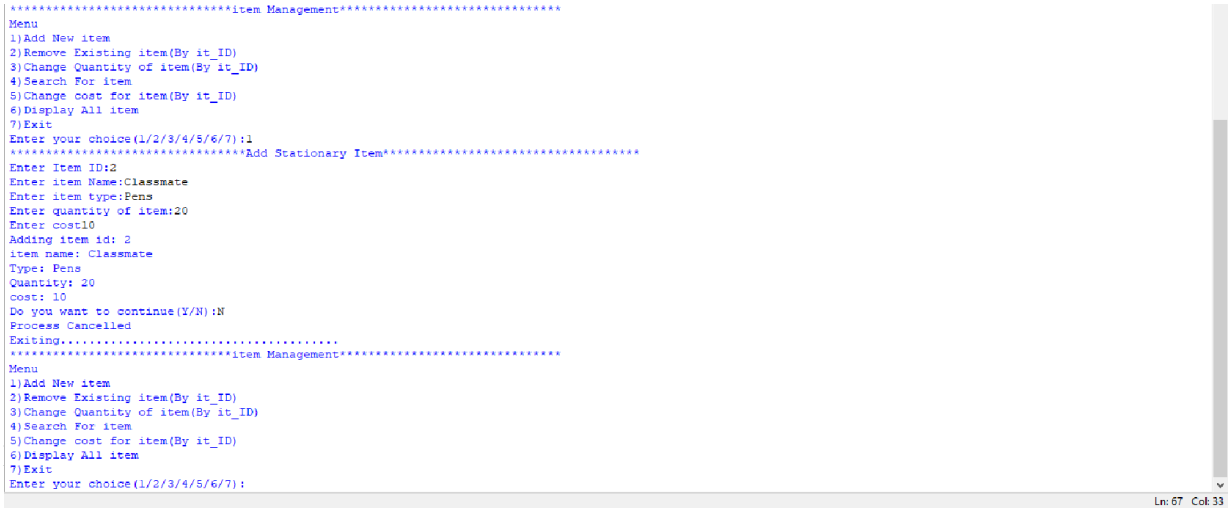
```
        elif a in ['3','3)']:
            tsearch()
        elif a in ['4','4)']:
            advsearch()
        elif a in ['5','5)']:
            main()
    def dispall():
        print("******************************All items*********************************")
        myc.execute("select * from item")
        rec=myc.fetchall()
        print('Item_ID'.ljust(15),'Name'.rjust(0),'Type'.rjust(15),'Quantity'.rjust(15),'cost'.rjust(14))
        for row in rec:

print(str(row[0]).rjust(4),str(row[1]).rjust(15),str(row[2]).rjust(15),str(row[3]).rjust(12),str(row[4]).rjust(15))
        main()
    print("*********************************item Management*****************************")
    print("Menu\n1)Add New item\n2)Remove Existing item(By it_ID)\n3)Change Quantity of item(By
it_ID)\n4)Search For item\n5)Change cost for item(By it_ID)\n6)Display All item\n7)Exit")
    a=(input("Enter your choice(1/2/3/4/5/6/7):"))
    if a in ['1','1)']:
        addst()
        main()
    elif a in ['2','2)']:
        remst()
        main()
    elif a in ['3','3)']:
        cquan()
        main()
    elif a in ['4','4)']:
        search()
        main()
    elif a in ['5','5)']:
        ccost()
        main()
    elif a in ['6','6)']:
        dispall()
        main()
    elif a in ['7','7)']:
        print("Exiting.....")
        con.close()
        sys.exit()

#main program
con=sq.connect(user='root',password='sunbeam',host='localhost')
myc=con.cursor()
ch=input("Do you want to create new tem Management Software(Y/N):")
if ch in ['Y','y']:
    myc.execute('create database if not exists item')
    myc.execute('use item')
    myc.execute('create table if not exists item(it_id int(4) primary key,name varchar(50),type varchar(15),quan
int(4),cost int(2))')
else:
    print('Exiting.....')
con=sq.connect(user='root',password='sunbeam',host='localhost',database='item')
myc=con.cursor()
```

main()

SAMPLE OUTPUT







```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):3
*****************************Change item Quantity*********************
Enter item Id:2
Enter New item Quantity:10
Changing item id: 2
item name: Apsara
Type: Pencil
Quantity: 20  To New Quantity: 10
cost: 3
Do you want to continue(Y/N):Y
Changed item id: 2
item name: Apsara
Type: Pencil
Quantity: 10
cost: 3
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 147 Col: 0

```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):2
*****************************Remove item*********************************
Enter item Id:2
Removing item id: 2
item name: Apsara
Type: Pencil
Quantity: 20
cost: 3
Do you want to continue(Y/N):N
Process Cancelled
Exiting.....................................
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 137 Col: 33

```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):3
*****************************Change item Quantity*********************
Enter item Id:2
Enter New item Quantity:5
Changing item id: 2
item name: Apsara
Type: Pencil
Quantity: 10  To New Quantity: 5
cost: 3
Do you want to continue(Y/N):N
Process Cancelled
Exiting.....................................
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 181 Col: 6

```
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):4
*********************************Search item**************************
Search by==>
1)item ID
2)Name
3)Type
4)Advanced Search
5)Exit To Menu
Enter choice to continue:3
Enter Type to search:Pen
Sort By==>
1)Quantity
2)cost
3)Do not sort
Enter choice to continue:2
1)Ascending
2)Decending
Enter choice to continue:2
Item_ID         Name            Type        Quantity          cost
   3           Trimax           Pen           50              55
   5       JC Whitener          Pen           10              10
   2           Apsara         Pencil          10               3
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 294 Col: 0

```
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):4
*********************************Search item**************************
Search by==>
1)item ID
2)Name
3)Type
4)Advanced Search
5)Exit To Menu
Enter choice to continue:2
Enter Name to search:Aps
Sort By==>
1)Quantity
2)cost
3)Do not sort
Enter choice to continue:1
1)Ascending
2)Decending
Enter choice to continue:1
Item_ID         Name            Type        Quantity          cost
   4           Apsara          Eraser          5               5
   2           Apsara          Pencil          10              3
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 262 Col: 6

```
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):5
*********************************Change item cost**************************
Enter item Id:3
Enter New item cost:60
Changing item id: 3
item name: Trimax
Type: Pen
Quantity: 50 cost: 55  to cost: 60
Do you want to continue(Y/N):Y
Changed item id: 3
item name: Trimax
Type: Pen
Quantity: 50
cost: 60
*******************************item Management*******************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 42 Col: 0

```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):5
*****************************Change item cost*************************
Enter item Id:1
Enter New item cost:50
Changing item id: 1
item name: Classmate
Type: Register
Quantity: 10 cost: 40  to cost: 50
Do you want to continue(Y/N):N
Process Cancelled
Exiting.....................................
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 88 Col: 6

```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):6
***************************All items*******************************
Item_ID          Name            Type        Quantity          cost
   1        Classmate        Register          10              40
   2          Apsara          Pencil          10               3
   3          Trimax             Pen          50              60
   4          Apsara          Eraser           5               5
   5      JC Whitener             Pen          10              10
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):
```

Ln: 32 Col: 6

```
*****************************item Management*****************************
Menu
1)Add New item
2)Remove Existing item(By it_ID)
3)Change Quantity of item(By it_ID)
4)Search For item
5)Change cost for item(By it_ID)
6)Display All item
7)Exit
Enter your choice(1/2/3/4/5/6/7):7
Exiting.....................................
>>>
```

Ln: 35 Col: 4

LIMITATIONS

1. This programme can not differentiate between fake or correct information being entered by the user. So, I must state that the user must verify the information.
2. Item name is designed to be at max 10 characters long. User is advised not to enter long names as to preserve the output table layout.
3. The item id which is being entered can be repeated, in this MYSQL error will occur as item id is primary key.
4. Any data type error committed during runtime will result in the program giving error and exiting.

BIBLIOGRAPHY

1.Class Notes by Pradumna Singh Sir

2.CBSE e-book

3.Computer Science with Python → by Sumita Arora

Thank

You