**PM SHRI KENDRIYA VIDYALAYA** **BERHAMPUR**

Session 2024-25

**A PROJECT REPORT ON**

**(ATL INVENTORY MANAGEMENT)**

**For**

 **CBSE 2025 EXAMINATION\_**

**[As a part of computer science course (083)]**

SUBMITTED BY : SUBMITTED TO:

SOUMYAJIT TRIPATHY Mr. S. K. MISRA

SUBRAT PADHI

SHIBASIS KUMAR SAHOO

 Class :XII A PGT (Computer Science)

**CERTIFICATE**

THIS IS TO CERTIFY THAT **(**SOUMYAJIT TRIPATHY**)** STUDYING IN CLASS XII A, HAS SATISFACTORILY COMPLETED PROJECT WITH THE TITLE **(ATL INVENTORY MANGEMENT)** UNDER THE GUIDANCE OF Mr. **SAROJ KANTA MISRA**, PGT (COMPUTER SCIENCE) DURING THE ACADEMIC YEAR 2024-25 IN PARTIAL FULFILLMENT OF “COMPUTER SCIENCE” PRACTICAL EXAMINATION OF CENTRAL BOARD OF SECONDARY EXAMINATION (CBSE)

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PRINCIPAL**

**Acknowledgement**

 **I warmly acknowledge the continuous encouragement and timely suggestions offered by our Principal Mr. SHIVAPRIYA DASH. I extend my hearty thanks for giving me the opportunity to make use of the facilities available in the campus to carry out the project successfully.**

 **I am highly indebted to Mr. SAROJ KANTA MISRA, (PGT Computer Science), for the constant supervision, providing necessary information and supporting in completing the project. I would like to express my gratitude towards them for their kind cooperation and encouragement.**

**Finally, I extend my gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.**

**NAME :–SOUMYAJIT TRIPATHY**

**CLASS :–XII A**

**ROLL.NO :–34**

**SIGN:-**

**CONTENTS**

1. **Introduction**

**2. System Implementation**

**2.1 The Hardware used:**

**2.2 The Software’s used:**

**3. System Design & Development**

**3.1 Python Coding**

**3.2 Database**

**3.3 Output Screen**

**4. References**

i **introduction**



❖ Python is a high-level language. It is a free and open- source language.

❖ It is an interpreted language, as Python programs are executed by an interpreter.

❖ Python programs are easy to understand as they have a clearly defined syntax and relatively simple structure.

❖ Python is case-sensitive. For example, NUMBER and number are not same in Python.

❖ Python is portable and platform independent, means it can run on various operating systems and hardware platforms.

❖ Python uses indentation for blocks and nested blocks.

**System Implementation**

**Hardware used:**

**While developing the software, the used hardware’s are: PC with Intel Core i3 processor having 4.00 GB RAM and other required devices.**

**Software used:**

**⮚ Microsoft Windows® 10 as Operating System.**

**⮚ Python IDLE as Front-end Development environment.**

**⮚ CSV Files as Back-end for storing the data.**

**⮚ MS-Word 2010 for documentation.**

**Python source code**

mport mysql.connector as m

from tabulate import tabulate as ta

def instructions1():

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Welcome to the ATL Management Software\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nThe **Atal Tinkering Lab Management System** is a Python-based application with MySQL integration designed to manage and track activities within Atal Tinkering Labs. It facilitates lab resource management, project tracking, student registration, and event scheduling, aiming to streamline operations and promote innovation in schools through hands-on learning.")

def instructions\_mentor():

    print("# Select 1 for Show Equipments\n# Select 2 for Items\n# Select 3 for Showing Equipments category Wise\n# Select 4 for Showing Products having more than 5 years warranty\n# Select 5 for Club Management\n# Select 0 for Back to Previous")

def issue\_equipments():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    # c2= db.cursor()

    try:

        name = input("Enter your name:")

        cl = int(input("Enter your class:"))

        sec = input("Enter your section:")

        no = input("Enter your mobile no. :")

        cy=0

        for i in d:

            if i[0] == pid:

                cy+=1

        if cy>0:

            c.execute("insert into issue values ('{}',{},'{}',{},{},'{}',{},'{}')".format(name,cl,sec,no,pid,pna,qnty,isd))

            print("Item issued")

        else:

            print("No Item Found of {} this ID".format(pid))

            return

        pid = int(input("Enter your product id :"))

        pna = input("Enter your product name :")

        isd = input("Enter your todays date :")

        qnty = int(input("Enter your quantity :"))

        c.execute("select pid from atl")

        d = c.fetchall()

        db.commit()

        db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

        c = db.cursor()

        c.execute("update atl set qty = qty-{} where productid = {}".format(qnty,pid))

    except ValueError:

        print("Invalid Input !!!")

        issue\_equipments()

    db.commit()

def return\_item():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    try:

        pid = int(input("Enter Product ID:"))

        try:

            c.execute("select product\_id,qty from issue where product\_id = {}".format(pid))

            f= c.fetchall()

            if len(f)==0:

                raise Exception

        except:

            print("No Item Issued of this Product ID {} ".format(pid))

            return

        no = int(input("Enter Product Quantity:"))

        name= input("Enter your name:")

        db.commit()

        c.execute("select name from issue")

        k = c.fetchall()

        u= ""

        for p in k:

            if p[0].lower()==name.lower():

                u+=p[0].lower()

        while True:

            for i in range(len(f)):

                if f[i][0]==pid and no<=f[i][1] and name.lower()==u.lower():

                    if f[i][1]>1:

                        c.execute("update issue set qty = qty-{} where name='{}'".format(no,name))

                        if no>f[i][1]:

                            print("The Quantity is not Valid !!!")

                    else:

                        c.execute("delete from issue where product\_id={}".format(pid))

                    c.execute("update atl set qty = qty+{}".format(no))

                    print("Returned Successfully!!!!\nData Updated In Main table !!!!")

                else:

                    print("Unmatch Data !!!!")

            break

    except ValueError:

        print("Invalid Input")

        return\_item()

    db.commit()

def show\_warranty\_start\_date():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select name,warranty\_start\_date from atl")

    d = c.fetchall()

    for i in d:

        print(ta(d,headers=['Product Name','Warranty Start Date'],tablefmt="psql"))

        # print(">Product Name:",i[0],"\n>Warranty Start Date:",i[1])

        print('\n')

    db.commit()

def show\_warranty\_end\_date():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select name,warranty\_end\_date from atl")

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=['Product Name',"Warranty End Date"],tablefmt="psql"))

        # print(">Product Name:",i[0],"\n>Warranty End Date:",i[1])

    print('\n')

    db.commit()

def warranty\_5\_years():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("Select category,name,warranty\_end\_date from atl where warranty\_end\_date>'{}'".format("2029-01-01"))

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=["Category",'Product Name',"Warranty End Date"],tablefmt="psql"))

        # print(">Category",i[0],"\n>Product Name",i[1],"\n>Warranty End Date",i[2])

    print("\n")

    db.commit()

def show\_all\_equipments():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute('select \* from atl')

    d = c.fetchall()

    print("All Item List --\n")

    # for i in d:

    print(ta(d,headers=["Slno","Category","Product Name","Quantity" ,"Price","Type","Product ID"],tablefmt="psql"))

        # print(">slno:",i[0],'\n>Category:',i[1],'\n>Product Name:',i[2],"\n>Quantity:",i[3],"\n>Price:",i[4],"\n>Type:",i[5],"\n>Product ID:",i[6])

    print("\n")

    db.commit()

def show\_issue\_table():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select \* from issue")

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=['Student Name',"class",'section','Mobile No','Product ID','Product Name','Quantity','Issue Date(YYYY-MM-DD)'],tablefmt="psql"))

        # print(">Name of the student:",i[0],"\n>class:",i[1],"\n>section:",i[2],"\n>Mobile No:",i[3],"\n>Product ID:",i[4],"\n>Product Name:",i[5],"\n>Quantity:",i[6],"\n>Issue Date(YYYY-MM-DD):",i[7])

    print("\n")

def add\_item():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select slno,productid from atl")

    f = c.fetchall()

    # for i in f:

    #     z = i[0]

    #     y = i[1]

    db.commit()

    try:

        n = int(input("Enter no of different products you want to add:"))

        for i in range(n):

            try:

                category = input("Enter category of product:")

                pn = input("Enter product name:")

                qty = int(input("Enter product quantity:"))

                pr = int(input("Enter product price:"))

                ty = input("Enter type of product:")

                # pr = int(input("Enter product ID:"))

                print("\*\* For Product Status Choose --\n# Select 0 for Working\n# Select 1 for Needs Repair\n")

                ch1 = int(input("Enter Choices:"))

                st = ""

                if ch1 == 0:

                      st+="Working"

                elif ch1==1:

                    st+="Needs Repair"

                else:

                    print("Enter Valid Input !!!")

                wsd = ""

                wed = ""

                print("\*\* For Warranty Date --\n# Select 0 for entering Starting Date of Warranty(YYYY-MM-DD)\n# Select 1 for entering Ending Date of Warranty(YYYY-MM-DD)\n")

                for i in range(2):

                    ch2= int(input("Enter your Choice: "))

                    if ch2 ==0:

                        d= int(input("Enter Start Date(YYYY-MM-DD): "))

                        mo= int(input("Enter Start Month: "))

                        y= int(input("Enter Start Year: "))

                        wsd += str(y)+"-"+str(mo)+"-"+str(d)

                    elif ch2==1:

                        d= int(input("Enter End Date(YYYY-MM-DD): "))

                        mo= int(input("Enter End Month: "))

                        y= int(input("Enter End Year: "))

                        wed += str(y)+"-"+str(mo)+"-"+str(d)

                    else:

                        print("Enter A Valid value !!!")

                c.execute("insert into atl values ({},'{}','{}',{},{},'{}',{},'{}','{}','{}')".format(f[-1][0]+1,category,pn,qty,pr,ty,f[-1][1]+1,st,wsd,wed))

                print("Product Successfully added !!!")

            except ValueError :

                print("Invalid Input !!!!")

                add\_item()

    except ValueError:

        print("Invalid Input!!!!")

        add\_item()

    db.commit()

def search\_by\_name():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    try:

        x = input("Enter Product Name:") #"%{}%"

    except:

        print("Enter a Valid Input!!")

        search\_by\_name()

    c.execute("select category,name,qty,productid,Product\_Status,warranty\_start\_date,warranty\_end\_date from atl where name like '%{}%'".format(x))

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=["Category",'Name','Quantity','Product ID','Product Status','Warranty Start Date(YYYY-MM-DD)','Warranty End Date(YYYY-MM-DD)'],tablefmt="psql"))

        # print(">Category",i[0],"\n>Name",i[1],"\n>Quantity",i[2],"\n>Product ID",i[3],"\n>Product Status",i[4],"\n>Warranty Start Date(YYYY-MM-DD)",i[5],"\n>Warranty End Date(YYYY-MM-DD)",i[6])

    print("\n")

    db.commit()

def low\_stock():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select \* from atl where qty<={} and type='{}'".format(4,"Consumable"))

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=["slno",'Category','Product Name','Quantity','Price','Type','Product ID'],tablefmt="psql"))

        # print(">slno:",i[0],'\n>Category:',i[1],'\n>Product Name:',i[2],"\n>Quantity:",i[3],"\n>Price:",i[4],"\n>Type:",i[5],"\n>Product ID:",i[6])

    print("\n")

    db.commit()

def delete\_item():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c= db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    try:

        n = int(input("Enter no of items you want to Delete:"))

        for i in range(n):

            pid = int(input("enter product ID:"))

            c.execute("delete from atl where productid={}".format(pid))

            print("Product Successfully Deleted !!!")

    except ValueError:

        print("Invalid Valid!!!!!")

        delete\_item()

    db.commit()

def product\_status():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select slno,name,Product\_Status from atl")

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=["slno",'Product Name','Product Status'],tablefmt="psql"))

        # print(">slno:",i[0],'\nProduct Name:',i[1],'\nProduct Status:',i[2])

    print("\n")

    db.commit()

def club\_instruction():

    print("# Select 1 for Robotics Club\n# Select 2 for Electronics Club\n# Select 0 for Back to Previous")

def club\_member\_registration():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    club\_instruction()

    try:

        ch = int(input("Enter your Choice:"))

        if ch==1:

            c.execute("select slno from robotics")

            d= c.fetchall()

            db.commit()

            n= input("Enter student Name: ")

            cl = int(input("Enter your class:"))

            sec= input("enter your section:")

            no= int(input("Enter student's Mobile Number:"))

            # em=""

            f = input("Enter student's email:")

            if f[0]!="@" and "." in f and f.count("@")==1 and f[-1]!=".":

                s = f.split(".")[-1]

                for i in s:

                    if not i.isdigit():

                        pass

            else:

                print("Enter valid Email!!!!")

                club\_member\_registration()

            c.execute("insert into robotics values ({},'{}',{},'{}',{},'{}')".format(d[-1][0]+1,n,cl,sec,no,f))

            print("Registration Completed !!!!")

            club\_instruction()

            db.commit()

        elif ch==2:

            c.execute("select slno from electronics")

            d= c.fetchall()

            db.commit()

            n= input("Enter student Name: ")

            cl = int(input("Enter your class:"))

            sec= input("enter your section:")

            no= int(input("Enter student's Mobile Number:"))

            # em=""

            f = input("Enter student's email:")

            if f[0]!="@" and "." in f and f.count("@")==1 and f[-1]!=".":

                s = f.split(".")[-1]

                for i in s:

                    if not i.isdigit():

                        pass

            else:

                print("Enter valid Email!!!!")

                club\_member\_registration()

            c.execute("insert into electronics values ({},'{}',{},'{}',{},'{}')".format(d[0][0]+1,n,cl,sec,no,f))

            print("Registration Completed !!!!")

            club\_instruction()

            db.commit()

        elif ch==0:

            return

    except ValueError:

        print("Invalid Input!!!!")

        club\_member\_registration()

def category\_searching():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    l = []

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("The categories:\n~Electronics Development\n~Internet Of Things\n~Robotics\n~Rapid Prototyping Tools\n~Mechanical Tools")

    try:

        x = input("Enter category name from above mentioned categories:")

        c.execute("select \* from atl where category like '%{}%'".format(x))

        d= c.fetchall()

        if len(d)>0:

            # for i in d:

                # l.append(i[1])

            print(ta(d,headers=["slno","Category",'Product Name','Quantity','Price','Type','Product ID'],tablefmt="psql"))

                # print(">slno:",i[0],'\n>Category:',i[1],'\n>Product Name:',i[2],"\n>Quantity:",i[3],"\n>Price:",i[4],"\n>Type:",i[5],"\n>Product ID:",i[6])

            print("\n")

        elif len(d)==0:

            print("No such category found")

    except ValueError:

        print("Invalid Input!!!")

        category\_searching()

    db.commit()

def need\_repair():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    c.execute("select name,Product\_Status where Product\_Status ='{}'".format("Needs\_Repair"))

    d = c.fetchall()

    # for i in d:

    print(ta(d,headers=['Product Name','Product Status'],tablefmt="psql"))

        # print(">Product Name:",i[0],"\n",">Product Status:",i[1],'\n')

#Join

def issue\_student\_name():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("# Select 0 for Product Id\n# Select 1 for Product Name")

    try:

        ch= int(input("Enter Choice:"))

        if ch ==0:

                try:

                    n = int(input("Enter Product ID:"))

                    c.execute("select product\_id,product\_name,issue.name,category,issue.qty,issue\_date from atl,issue where atl.productid=issue.product\_id and product\_id = {}".format(n))

                    f = c.fetchall()

                    if len(f)==0:

                        print("No Product ID Found!!!")

                        issue\_student\_name()

                    else:

                        # for i in f:

                        print(ta(f,headers=["Product ID","Product Name","Student Name","Category","Issue Quantity",'Issue Date(YYYY-MM-DD)'],tablefmt="psql"))

                            # print("# Product ID: ",i[0],"\n# Product Name: ",i[1],"\n# Student Name",i[2],"\n# Category: ",i[3],"\n# Issue Quantity: ",i[4],"\n# Issue Date(YYYY-MM-DD): ",i[5],"\n")

                except ValueError:

                    print("Enter valid Input !!!!")

                    issue\_student\_name()

        elif ch==1:

            try:

                n= input("Enter Product name: ")

                c.execute("select product\_id,product\_name,issue.name,category,issue.qty,issue\_date from atl,issue where atl.productid=issue.product\_id and product\_name = '{}'".format(n))

                f = c.fetchall()

                if len(f)==0:

                        print("No Product ID Found!!!")

                        issue\_student\_name()

                else:

                        # for i in f:

                        print(ta(f,headers=["Product ID","Product Name","Student Name","Category","Issue Quantity",'Issue Date(YYYY-MM-DD)'],tablefmt="psql"))

                            # print("# Product ID: ",i[0],"\n# Product Name: ",i[1],"\n# Student Name: ",i[2],"\n# Category: ",i[3],"\n# Issue Quantity: ",i[4],"\n# Issue Date(YYYY-MM-DD): ",i[5],"\n")

            except ValueError:

                print("Enter valid Input !!!!")

                issue\_student\_name()

        else:

            print("Enter Valid Input!!!!")

            issue\_student\_name()

    except ValueError:

        print("Enter valid Input !!!!")

        issue\_student\_name()

def club\_update():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    x = int(input("Enter slno of the member:"))

    print("# Select 1 for Robotics Club\n# Select 2 for Electronics Club\n# Select 0 for Back to Previous")

    try:

        t = int(input("Enter your choice: "))

        if t ==1 :

            c.execute("select slno from robotics")

            d = c.fetchall()

            cw=0

            db.commit()

            for i in d:

                if i[0]==x:

                    cw+=1

            if cw>0:

                print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

                print("# Select 1 for Updating Name")

                print("# Select 2 for Updating class")

                print("# Select 3 for Updating Section")

                print("# Select 4 for Updating Mobile Number")

                print("# Select 5 for Updating email")

                print("# Select 0 for Back to Previous")

                y = int(input("Enter your choice: "))

                if y == 1:

                    z = input("Enter value:")

                    c.execute("update robotics set name = '{}' where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 2:

                    z = input("Enter value:")

                    c.execute("update robotics set class = {} where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 3:

                    z = input("Enter value:")

                    c.execute("update robotics set section = '{}' where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 4:

                    z = input("Enter value:")

                    c.execute("update robotics set mobile = {} where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 5:

                    f = input("Enter value:")

                    if f[0]!="@" and "." in f and f.count("@")==1 and f[-1]!=".":

                        s = f.split(".")[-1]

                        for i in s:

                            if not i.isdigit():

                                pass

                    else:

                        print("Enter valid Email!!!!")

                        club\_update()

                    c.execute("update robotics set email = '{}' where slno ={}".format(f,x))

                    db.commit()

                    print("Updated !!!!")

            else:

                print("No Slno found !!!!")

        elif t ==2:

            c.execute("select slno from electronics")

            d = c.fetchall()

            cw=0

            db.commit()

            for i in d:

                if i[0]==x:

                    cw+=1

            if cw>0:

                print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

                print("# Select 1 for Updating Name")

                print("# Select 2 for Updating class")

                print("# Select 3 for Updating Section")

                print("# Select 4 for Updating Mobile Number")

                print("# Select 5 for Updating email")

                y = int(input("Enter your choice: "))

                if y == 1:

                    z = input("Enter value:")

                    c.execute("update electronics set name = '{}' where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 2:

                    z = input("Enter value:")

                    c.execute("update electronics set class = {} where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 3:

                    z = input("Enter value:")

                    c.execute("update electronics set section = '{}' where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 4:

                    z = input("Enter value:")

                    c.execute("update electronics set mobile = {} where slno ={}".format(z,x))

                    db.commit()

                    print("Updated !!!!")

                elif y == 5:

                    f = input("Enter value:")

                    if f[0]!="@" and "." in f and f.count("@")==1 and f[-1]!=".":

                        s = f.split(".")[-1]

                        for i in s:

                            if not i.isdigit():

                                pass

                    else:

                        print("Enter valid Email!!!!")

                        club\_update()

                    c.execute("update electronics set email = '{}' where slno ={}".format(f,x))

                    db.commit()

                    print("Updated !!!!")

            else:

                print("No Slno Found!!!!")

        elif t==0:

            return

    except ValueError:

        print("Invalid Input !!!")

        club\_update()

def club\_members():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("# Select 1 for Robotics Club\n# Select 2 for Electronics Club\n# Select 0 for Back to Previous")

    try:

        x = int(input("Enter your choice: "))

        if x==1:

            c.execute("select \* from robotics")

            d= c.fetchall()

            db.commit()

            # for i in d:

            print(ta(d,headers=['slno','Member Name','Class','Section','Mobile No','Email'],tablefmt="psql"))

                # print("\n>slno:",i[0],"\n>Member Name:",i[1],"\n>Class:",i[2],"\n>Section:",i[3],"\n>Mobile No:",i[4],"\n>Email",i[5],"\n")

        elif x==2:

            c.execute("select \* from electronics")

            d= c.fetchall()

            db.commit()

            # for i in d:

            print(ta(d,headers=['slno','Member Name','Class','Section','Mobile No','Email'],tablefmt="psql"))

                # print("\n>slno:",i[0],"\n>Member Name:",i[1],"\n>Class:",i[2],"\n>Section:",i[3],"\n>Mobile No:",i[4],"\n>Email",i[5],"\n")

        elif x==0:

            return

        else:

            print("Enter A Valid Value Try Again!!!!")

            club\_projects()

    except ValueError:

        print("Invalid Input !!!! ")

        club\_members()

def item():

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("\n# Select 1 for Adding Item\n# Select 2 for Deleting An Item\n# Select 3 for showing Low Stock\n# Select 4 for showing Product Status\n# Select 5 Updating Equipments\n# Select 6 for Showing Items which needs Repair\n# Select 7 for Showing Warranty Start Date\n# Select 8 for Showing Warranty End Date\n# Select 0 for Back to Previous")

    try:

        x = int(input("Enter Your Choices: "))

        if x == 1:

            add\_item()

            item()

        elif x==2:

            delete\_item()

            item()

        elif x==3:

            low\_stock()

            item()

        elif x==4:

            product\_status()

            item()

        elif x==5:

            update\_equipments()

            item()

        elif x==6:

            need\_repair()

            item()

        elif x==7:

            show\_warranty\_start\_date()

            item()

        elif x==8:

            show\_warranty\_end\_date()

        elif x==9:

            show\_issue\_table()

        elif x==0:

            return

        else:

            print("Enter A Valid Value!!!!")

            item()

    except ValueError:

        print("Invalid Input !!!!")

        item()

def choices\_mentor():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    while True:

        try:

            ch =int(input("Enter your choice:"))

            if ch == 1:

                show\_all\_equipments()

                instructions\_mentor()

            elif ch == 2:

                item()

                instructions\_mentor()

            elif ch==3:

                category\_searching()

                instructions\_mentor()

            elif ch==4:

                warranty\_5\_years()

                instructions\_mentor()

            elif ch==5:

                club()

                instructions\_mentor()

            elif ch == 0:

                # print("Thank You for using our Software!!")

                print("Please Login Again ----")

                return

            else:

                print("Enter A Valid Value!!!!")

                instructions\_mentor()

                choices\_mentor()

        except ValueError:

            print("Invalid Input !!!")

            instructions\_mentor()

            choices\_mentor()

    db.commit()

def choices\_student():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    while True:

        try:

            print("\n# Select 1 for Showing All the Equipments\n# Select 2 for Item\n# Select 3 for Back to Previous\n# Select 0 for Exit")

            ch =int(input("Enter your choice:"))

            if ch==1:

                show\_all\_equipments()

                # instructions\_student()

            elif ch==2:

                item\_student()

                # instructions\_student()

            elif ch==3:

                login()

            elif ch==0:

                print("Please Login Again ----")

                return

            else:

                print("Enter Valid Input!!!!")

                choices\_student()

        except ValueError:

            print("Invalid Input!!!!")

            choices\_student()

    db.commit()

def login():

    db = m.connect(host = "localhost",user = "root",password = "admin",database = "cs24")

    c = db.cursor()

    v = True

    while True:

        try:

            print("# Select 1 for Student Login")

            print("# Select 2 for Mentor Login")

            print("# Select 0 Exit")

            ch = int(input("Enter choice:"))

            if ch==1:

                user= input("Enter Username:")

                pw= int(input("Enter password:"))

                c.execute("select \* from student\_login")

                d = c.fetchall()

                o = len(d)

                z = 0

                for i in d:

                    if i[0]==user.lower() and i[1]==pw:

                        print("Login successful")

                        instructions1()

                        # instructions\_student()

                        choices\_student()

                        # v = False

                        break

                    else:

                        z +=1

                if z == o:

                    print("No user and password found")

                db.commit()

            elif ch == 2:

                user= input("Enter Username:")

                pw= int(input("Enter password:"))

                c.execute("select \* from mentor\_login")

                d = c.fetchall()

                t = len(d)

                r = 0

                for i in d:

                    if i[0]==user.lower() and i[1]==pw:

                        print("Login successful")

                        instructions1()

                        instructions\_mentor()

                        choices\_mentor()

                        v= False

                    else:

                        r +=1

                if r == t:

                    print("No user and password found")

                # break

            elif ch==0:

                print("Thank You for using our Software!!")

                v= False

                break

            else:

                print("Enter A Valid Value !!!")

        except ValueError:

            print("Invalid Input!!!")

            login()

login()

**OUTPUT**

 **Main Menu**

****





**# Search Product By Name (LIKE)**



**MENTOR LOGIN:** 



**# Club Project Members(Foreign Key)**



**# Club Member Searching By Section**





**# Showing Warranty Dates of Products**



**# Updating an Item**



 **DATABASES**

 **MENTOR LOGIN**



 **STUDENT LOGIN**



 **MAIN TABLE**

 **ISSUE TABLE**



 **CLUB MEMBERS**

 

 **ROBOTICS CLUB**



 **ELECTRONICS CLUB**



 **PROJECT TABLE**

 **REFERENCE**

In order to work on this project titled –ATL INVENTORY MANAGEMENT, the following books and websites are

referred by me during the various phases of

development of the project.

1) Computer Science (Sumita arora Class XI and XII)

2) www.youtube.com

3) www.python.com

Other than the above-mentioned books, the suggestions

and supervision of my teacher and my class experience

also helped me to develop this software project.

 **THANK YOU**