

## SPRINT 4

Team ID	PNT2022TMID08788
Project Name	Inventory Management for Retailers
Maximum Marks	4 Marks

```
from flask import Flask, render_template, request, redirect, url_for
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

app = Flask(__name__)

conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io
90108kqblod8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServiceCertif
icate=DigiCertGlobalRootCA.crt;UID=knn47340;PWD=A67srRSfZM2UneYF",'','')

@app.route("/")
def index():
    return render_template('index.html')

@app.route('/aboutus')
def login():
    return render_template('aboutus.html')

@app.route("/signin")
def blog():
    return render_template('signin.html')

@app.route('/signup')
def signup():
    return render_template('signup.html')

@app.route('/addProduct')
def addProduct():
    return render_template('addProduct.html')

@app.route('/updateProduct')
def updateProduct():
    return render_template('updateProduct.html')
```

```

@app.route('/doUpdateProduct',methods = ['POST', 'GET'])
def doUpdateProduct():
    if request.method == 'POST':
        try:
            print("-----")
            pName=request.form['pName']
            pQuantity=request.form['pQuantity']
            insert_sql = "UPDATE inventory SET quantity=? WHERE name=?"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, pQuantity)
            ibm_db.bind_param(prepare_stmt, 2, pName)
            print(prepare_stmt)
            ibm_db.execute(prepare_stmt)
            return render_template('redirectTodashboard.html',msg="Successfully
updated the product")
        except:
            print("error")

@app.route('/doAddProduct',methods = ['POST', 'GET'])
def doAddProduct():
    if request.method == 'POST':
        try:
            pName=request.form['pName']
            pCategory=request.form['pCategory']
            pQuantity=request.form['pQuantity']
            insert_sql = "INSERT INTO inventory VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, pName)
            ibm_db.bind_param(prepare_stmt, 2, pCategory)
            ibm_db.bind_param(prepare_stmt, 3, pQuantity)
            ibm_db.execute(prepare_stmt)
            return render_template('redirectTodashboard.html',msg="Successfully added
the product")
        except:
            print("error")

@app.route('/dashboard')
def dashboard():
    sql="SELECT * FROM inventory ORDER BY category"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    data=ibm_db.fetch_assoc(stmt)
    datas=[]
    while data!=False:
        # if(data['QUANTITY']<10):
        #     sendmail("Your Inventory with product name "+data['NAME']+" is running out

```

```

of stock")
    datas.append(data)
    data=ibm_db.fetch_assoc(stmt)
    return render_template('dashboard.html',data_list=datas)

@app.route('/doSignin',methods = ['POST', 'GET'])
def doSignin():
    if request.method == 'POST':

        email = request.form['email']
        password = request.form['password']

        sql = "SELECT * FROM users WHERE email =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('index.html')
        else:
            return render_template('signin.html', msg="Invalid credentials")

@app.route('/doSignup',methods = ['POST', 'GET'])
def doSignup():
    if request.method == 'POST':

        name = request.form['name']
        email = request.form['email']
        password = request.form['password']
        cpassword = request.form['cpassword']

        if password!=cpassword:
            return render_template('signup.html', msg="Password don't match")

        sql = "SELECT * FROM users WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('signup.html', msg="Email already exists")
        else:

```

```

        insert_sql = "INSERT INTO users (name,email,password) VALUES (?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, name)
        ibm_db.bind_param(prepare_stmt, 2, email)
        ibm_db.bind_param(prepare_stmt, 3, password)
        ibm_db.execute(prepare_stmt)

    return render_template('signin.html', msg="You have successfully signed up,You
can login now")

def sendmail(msg):
    message = Mail(
        from_email='malleshdon22@gmail.com',
        to_emails='malleshwar22@gmail.com',
        subject='Warning Product running out of stock!',
        html_content=msg)
    try:
        sg =
SendGridAPIClient(os.environ.get('SG.zhP35A2VTpeMG1zmamkXsg.Z66fPFzxOojbdRut9L0wBH
Bk6QCw0CZ5Yw4mpagqQ28'))
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        print(e.message)

app.run()

```

Applications Places System IBM Db2 on Cloud - Brave

IBM Watson Service Pa IBM Watson Assistant IBM Db2 on Cloud download json of watsc

bs2jpcul0apon0jufi80lite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fc52ee092929844739\_

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

KNN47340.INVENTORY Back

Export to CSV

NAME	CATEGORY	QUANTITY
60W vooc adapter	Accessories	5
Fan	Appliances	12
Grinder	Appliances	20
Micro USB Cable	Accessories	10
USB Type-C	Electronics	35
iPhone 13	Electronics	60
iPhone 13 pro	Electronics	45
iPhone 14	Electronics	40
iPhone 14 pro	Electronics	10
macbook intel	Electronics	20
macbook m1	Electronics	25
mixer	Appliances	10

app.py - project - VSCo... IBM Db2 on Cloud - Bra... project vokoscreenNG-2022-1... Parrot Terminal

Applications Places System Dashboard - Brave

About Us Google Docs: Online Sprint\_1.docx - Goog Sprint\_1.docx - Goog sprin\_3.docx - Google WhatsApp IBM Cloud Dashboard

127.0.0.1:5000/dashboard

Inventory Management About us Dashboard Sign out

PRODUCTNAME	CATEGORY	AVAILABLE QUANTITY
Micro USB Cable	Accessories	10
60W vooc adapter	Accessories	5
pen drive 32GB	Accessories	25
mixer	Appliances	10
Grinder	Appliances	20
Fan	Appliances	12
iPhone 14	Electronics	40
iPhone 14 pro	Electronics	10
iPhone 13	Electronics	60
iPhone 13 pro	Electronics	45
macbook intel	Electronics	20
macbook m1	Electronics	25
USB Type-C	Electronics	35

Add Product Update Product

Dashboard - Brave app.py - Source - ... Sprint\_1.docx - Li... 3 test.py - ibm - VS... Source users\_db2.png Screenshot at 202...

