SPRINT 4

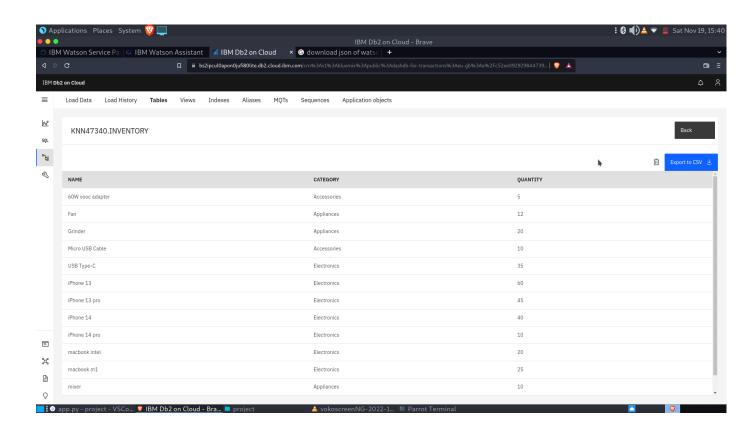
Team ID	PNT2022TMID08788
Project Name	Inventory Management for Retailers
Maximum Marks	4 Marks

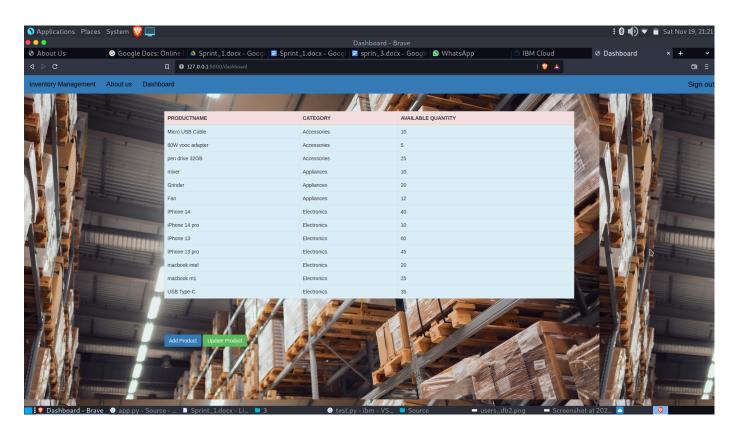
```
from flask import Flask, render_template, request, redirect, url_for
import ibm db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
app = Flask(__name__)
conn =
ibm db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io
90108kqb1od81cg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServiceCertif
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')
```

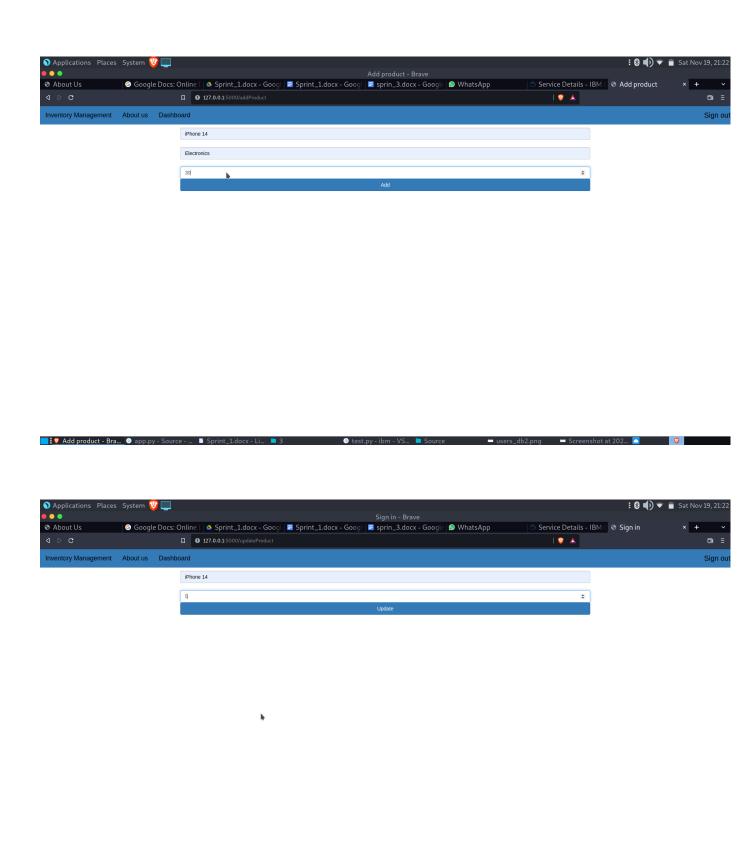
```
@app.route('/doUpdateProduct',methods = ['POST', 'GET'])
def doUpdateProduct():
  if request.method == 'POST':
    try:
        print("----")
        pName=request.form['pName']
        pQuantity=request.form['pQuantity']
        insert sql = "UPDATE inventory SET quantity=? WHERE name=?"
        prep stmt = ibm db.prepare(conn, insert sql)
        ibm_db.bind_param(prep_stmt, 1, pQuantity)
        ibm_db.bind_param(prep_stmt, 2, pName)
        print(prep_stmt)
        ibm db.execute(prep 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(prep stmt, 1, pName)
      ibm_db.bind_param(prep_stmt, 2, pCategory)
      ibm_db.bind_param(prep_stmt, 3, pQuantity)
      ibm db.execute(prep 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['OUANTITY']<10):</pre>
       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')
      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(prep stmt, 1, name)
     ibm_db.bind_param(prep_stmt, 2, email)
     ibm db.bind param(prep stmt, 3, password)
     ibm_db.execute(prep_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
{\tt Bk6QCw0CZ5Yw4mpagqQ28'))}
     response = sg.send(message)
     print(response.status code)
     print(response.body)
    print(response.headers)
except Exception as e:
    print(e.message)
app.run()
```







Sign in - Brave o app.py - Source - ... Sprint_1.docx - Li... ■ 3