ASSIGNMENT 3

A.ANCY RENISHA

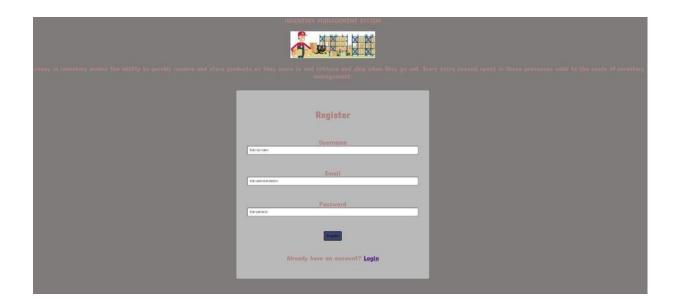
VV COLLEGE OF ENGINEERING

III- YEAR CSE

(TEAM LEADER)

INVENTORY MANAGEMENT SYSTEM

	Login	
Username		
Password		
	Login	



Docker image to the IBM container registry

```
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=54a2f15b-5c0f-46df-89
print("connected")
```

```
@app.route('/')
idef home():
    return render_template('index.html')
Qapp.route('/home')
def home1():
    return render_template('index.html')
@app.route('/about')
def about():
    return render_template('index.html')
@app.route('/meme')
def mem():
    return render_template('meme.html')
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/register')
|def register():
    return render_template('register.html')
@app.route('/logout')
|def logout():
    return render_template('index.html')
```

```
@app.route("/reg", methods=['POST', 'GET'])
Idef signup():
    msg = ''
    if request.method == 'POST':
        name = request.form["name"]
        email = request.form["email"]
        password = request.form["password"]
        sql = "SELECT * FROM REGISTER WHERE name= ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            return render_template('login.html', error=True)
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = "Invalid Email Address!"
            insert_sql = "INSERT INTO REGISTER 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)
            msg = "You have successfully registered !"
    return render_template('login.html', msg=msg)
```

```
@app.route("/log", methods=['POST', 'GET'])
def login1():
   if request.method == "POST":
        email = request.form["email"]
       password = request.form["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)
        print(account)
        if account:
           session['Loggedin'] = True
            session['id'] = account['EMAIL']
            session['email'] = account['EMAIL']
           return render_template('meme.html')
           msg = "Incorrect Email/password"
           return render_template('login.html', msg=msg)
        return render_template('login.html')
```

Output



Docker container using kubernetes



