

## Final Deliverables Report

<b>Team ID</b>	PNT2022TMID21037
<b>Project Name</b>	Inventory Management System for Retailers

### Team members and their Contribution:

<b>Name</b>	<b>Roll no</b>	<b>Contribution</b>
Arun Kumar	714019104015	Frontend – 5 Pages, Deployment of using docker and Kubernetes.
Haneef	714019104030	Frontend – 5 Pages, Documentation
Hareesh Kumaran	714019104031	Frontend – 4 Pages, Documentation
Kaushik	714019104045	Backend Fully , Integration of IBM Cloud, Deployment of using docker and Kubernetes.

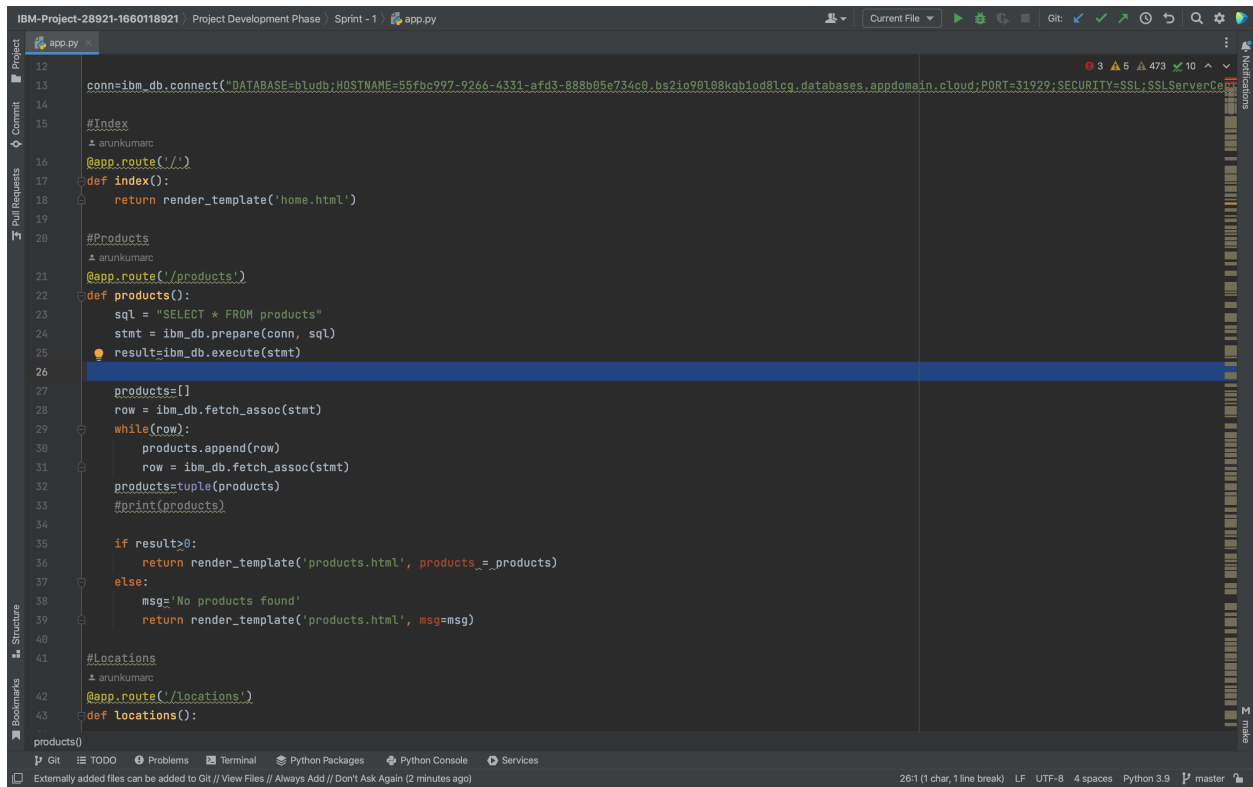
### Introduction:

1. Sprint 1 -Backend
2. Sprint 2 -Frontend
3. Sprint 3 -IBM cloud integration
4. Sprint 4 -Docker and Kubernetes

## Sprint 1 - Backend:

All the routes to each page and APIs are created.

Example, (For Products page)

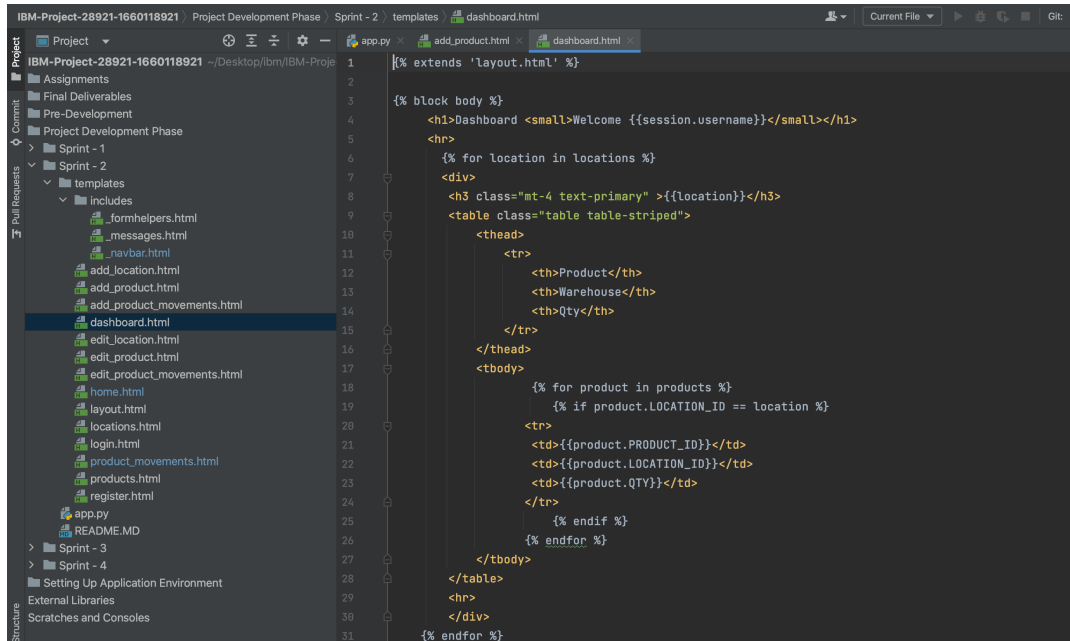


```
IBM-Project-28921-1660118921 | Project Development Phase | Sprint - 1 | app.py
12
13 conn=ibm_db.connect('DATABASE=b1udb;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lq.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLServerCe
14
15 #Index
16 anunkumar
17 @app.route('/')
18 def index():
19     return render_template('home.html')
20
21 #Products
22 anunkumar
23 @app.route('/products')
24 def products():
25     sql = "SELECT * FROM products"
26     stmt = ibm_db.prepare(conn, sql)
27     result=ibm_db.execute(stmt)
28
29     products=[]
30     row = ibm_db.fetch_assoc(stmt)
31     while(row):
32         products.append(row)
33         row = ibm_db.fetch_assoc(stmt)
34     products=tuple(products)
35     #print(products)
36
37     if result>0:
38         return render_template('products.html', products=_products)
39     else:
40         msg='No products found'
41         return render_template('products.html', msg=msg)
42
43 #Locations
44 anunkumar
45 @app.route('/locations')
46 def locations():
47     products()
48
49 products()
50
51 Git | TODO | Problems | Terminal | Python Packages | Python Console | Services
52 Externally added files can be added to Git // View Files // Always Add // Don't Ask Again (2 minutes ago)
53 26:1 (1 char, 1 line break) | LF | UTF-8 | 4 spaces | Python 3.9 | master
```

## Sprint 2 - Frontend:

The frontend is written using HTML, CSS (using Bootstrap) and JavaScript for all the pages to which the routes created in Sprint 1.

For Example, (The Hierarchy of different pages and the code for login page)

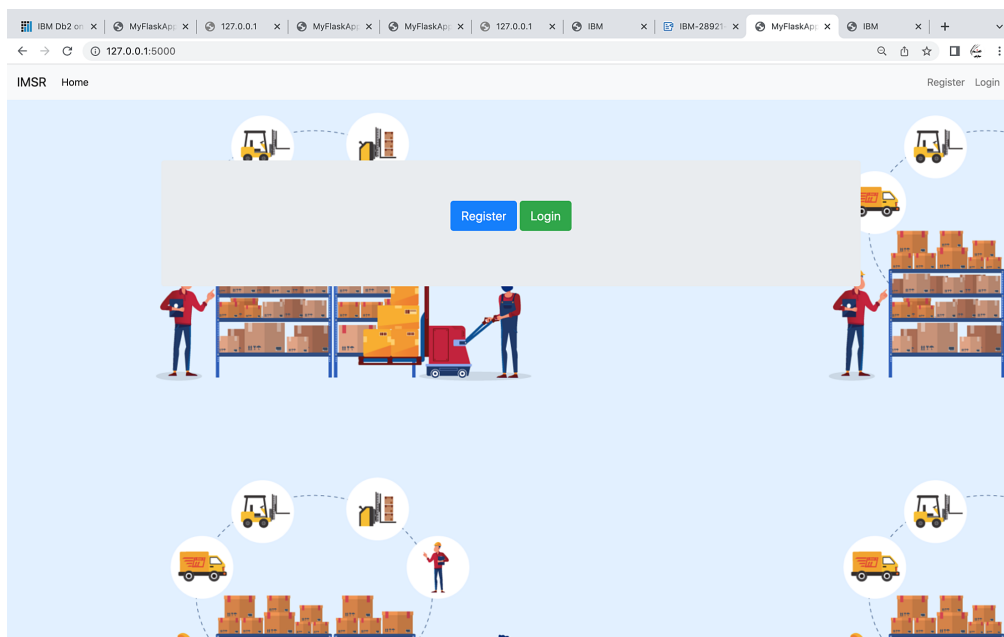


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'IBM-Project-28921-1660118921' with a 'templates' folder containing 'dashboard.html'. The code editor shows the content of 'dashboard.html' which extends 'layout.html' and displays a dashboard with a table of products.

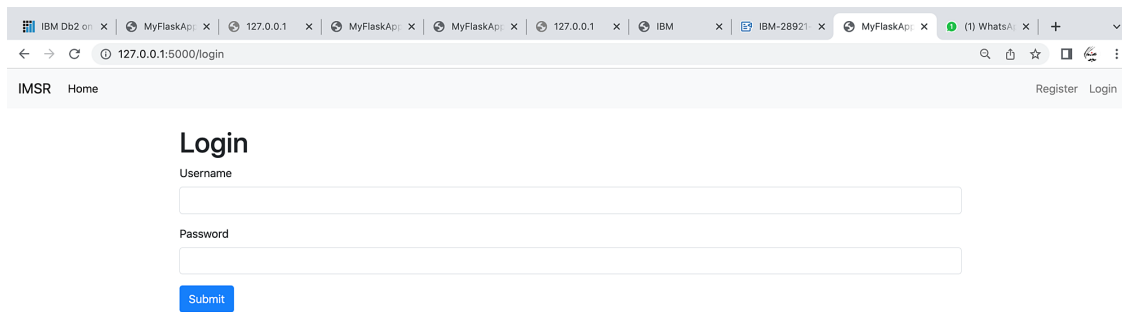
```
1 {% extends 'layout.html' %}
2
3 {% block body %}
4 <h1>Dashboard <small>Welcome {{session.username}}</small></h1>
5 <hr>
6 {% for location in locations %}
7 <div>
8 <h3 class="mt-4 text-primary" >{{location}}</h3>
9 <table class="table table-striped">
10 <thead>
11 <tr>
12 <th>Product</th>
13 <th>Warehouse</th>
14 <th>Qty</th>
15 </tr>
16 </thead>
17 <tbody>
18 {% for product in products %}
19 {% if product.LOCATION_ID == location %}
20 <tr>
21 <td>{{product.PRODUCT_ID}}</td>
22 <td>{{product.LOCATION_ID}}</td>
23 <td>{{product.QTY}}</td>
24 </tr>
25 {% endif %}
26 {% endfor %}
27 </tbody>
28 </table>
29 <hr>
30 </div>
31 {% endfor %}
```

Sample FrontEnd Pages,

Home Page,



## Login page,



The screenshot shows a web browser with multiple tabs. The active tab is titled "MyFlaskApp". The address bar shows the URL "127.0.0.1:5000/login". The page has a header with "IMSR" and "Home" on the left, and "Register" and "Login" on the right. The main content area is titled "Login" and contains two input fields: "Username" and "Password". Below the "Password" field is a blue "Submit" button.

IBM Db2 on X | MyFlaskApp X | 127.0.0.1 X | MyFlaskApp X | MyFlaskApp X | 127.0.0.1 X | IBM X | IBM-28921 X | MyFlaskApp X | (1) WhatsA X | +

← → ↻ 127.0.0.1:5000/login 🔍 📄 ☆ 🖨️ ⋮

IMSR Home Register Login

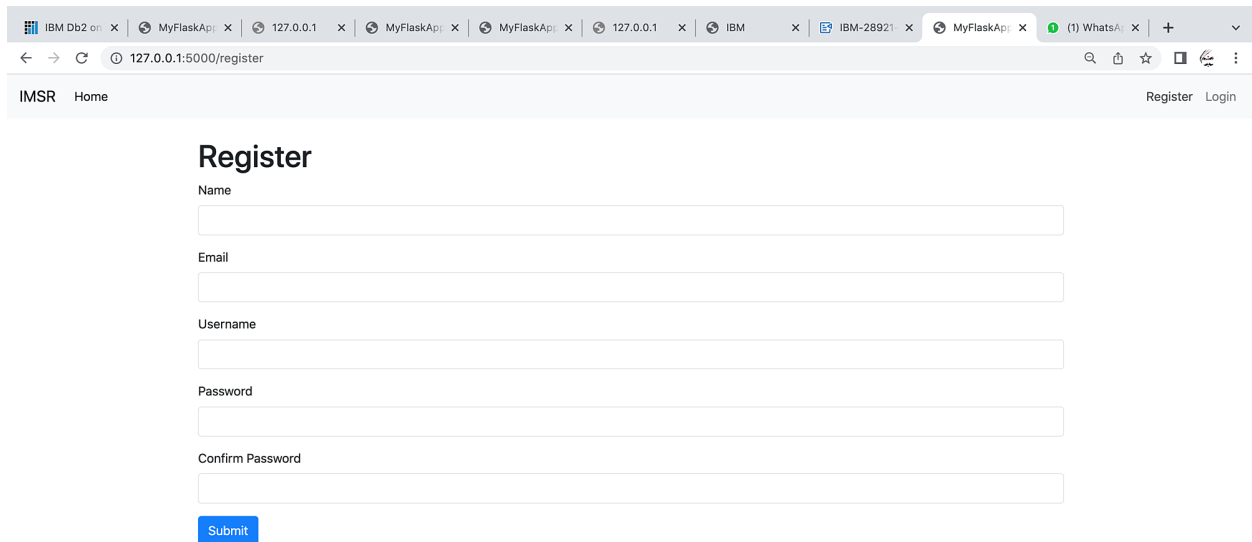
## Login

Username

Password

Submit

## Register,



The screenshot shows a web browser with multiple tabs. The active tab is titled "MyFlaskApp". The address bar shows the URL "127.0.0.1:5000/register". The page has a header with "IMSR" and "Home" on the left, and "Register" and "Login" on the right. The main content area is titled "Register" and contains five input fields: "Name", "Email", "Username", "Password", and "Confirm Password". Below the "Confirm Password" field is a blue "Submit" button.

IBM Db2 on X | MyFlaskApp X | 127.0.0.1 X | MyFlaskApp X | MyFlaskApp X | 127.0.0.1 X | IBM X | IBM-28921 X | MyFlaskApp X | (1) WhatsA X | +

← → ↻ 127.0.0.1:5000/register 🔍 📄 ☆ 🖨️ ⋮

IMSR Home Register Login

## Register

Name

Email

Username

Password

Confirm Password

Submit

products page,

IBM Db2 on IBMMyFlaskApp127.0.0.1MyFlaskApp1MyFlaskApp1127.0.0.1IBMIBM-28921MyFlaskApp1(1) WhatsApp

127.0.0.1:5000/products

LogoutDashboard

IMSRHomeProductsLocationProduct Movements

Products

Add Product

Product ID	Product Cost	Product Quantity		
biscut	20	113	Edit	Delete

product movents page,

IBM Db2 on IBMMyFlaskApp1127.0.0.1MyFlaskApp1MyFlaskApp1127.0.0.1IBMIBM-28921MyFlaskApp1WhatsApp

127.0.0.1:5000/product\_movements

LogoutDashboard

IMSRHomeProductsLocationProduct Movements

Product Movements

Add Product Movements

Movement ID	Time	From Location	To Location	Product ID	Quantity	
2		Main Inventory	kerala	biscut	20	Delete

## Sprint 3 - IBM Cloud Integration

### IBM Cloud Integration:

5 tables created for our project,

The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The main content area is divided into two panels. The left panel, titled 'Schemas', contains a table with the following data:

Name	Type	Tables
BXF26731	User	5

The right panel, titled 'Tables', contains a table with the following data:

Name	Schema	Properties
LOCATIONS	BXF26731	...
PRODUCTMOVEMENTS	BXF26731	...
PRODUCTS	BXF26731	...
PRODUCT_BALANCE	BXF26731	...
USERS	BXF26731	...

At the bottom of each panel, there are status indicators: 'Total: 1, selected: 1' for Schemas and 'Total: 5, selected: 0' for Tables.

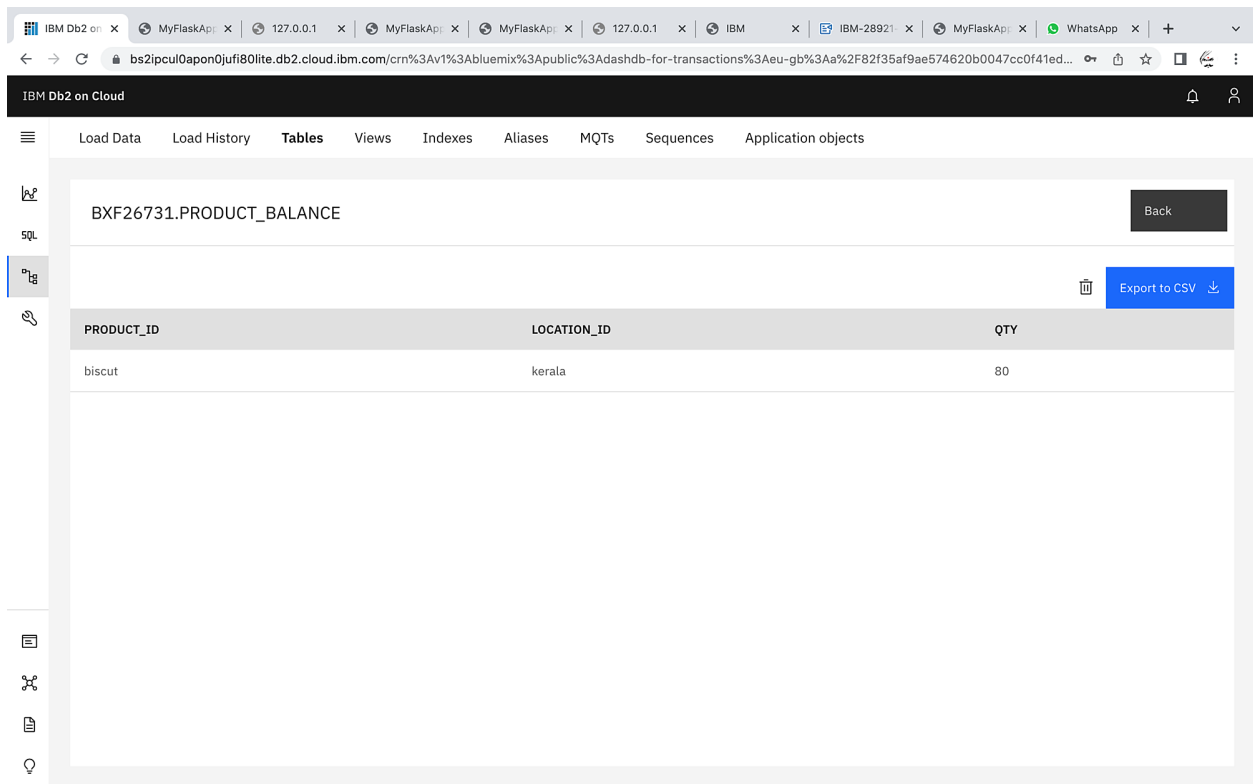
### Schema of the particular table (For Example, Product\_Balance)

The screenshot shows the IBM Db2 on Cloud web interface with the 'Table definition' panel selected for the 'PRODUCT\_BALANCE' table. The table definition panel displays the following columns:

Name	Data type	Nullable	Length	Scale
PRODUCT_ID	LONG VARCHAR	Y	32700	0
LOCATION_ID	LONG VARCHAR	Y	32700	0
QTY	INTEGER	Y		0

The left panel, titled 'Tables', shows the same list of tables as in the previous screenshot. The 'View data' button is visible at the bottom of the table definition panel.

## Data of a particular table (For Example, Product\_Balance)



The screenshot displays the IBM Db2 on Cloud interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The main content area shows the table BXF26731.PRODUCT\_BALANCE. A 'Back' button is located in the top right corner of the table view. Below the table name, there is a table with the following data:

PRODUCT_ID	LOCATION_ID	QTY
biscut	kerala	80

On the right side of the table, there is an 'Export to CSV' button with a download icon.

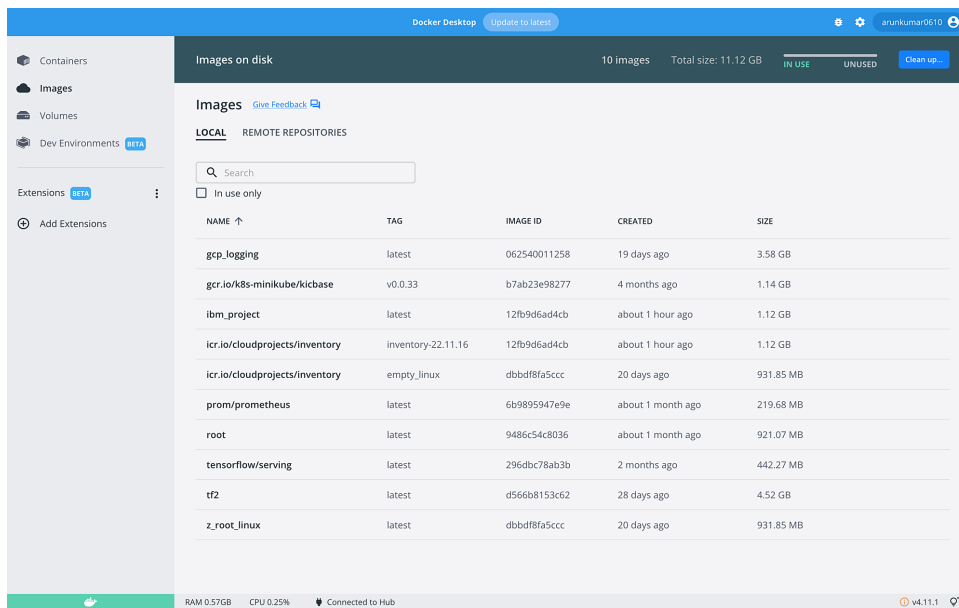
Code for Connection of IBM Database,

**Note:** `DigiCertGlobalRootCA.crt` should be downloaded and configured within the project folder.

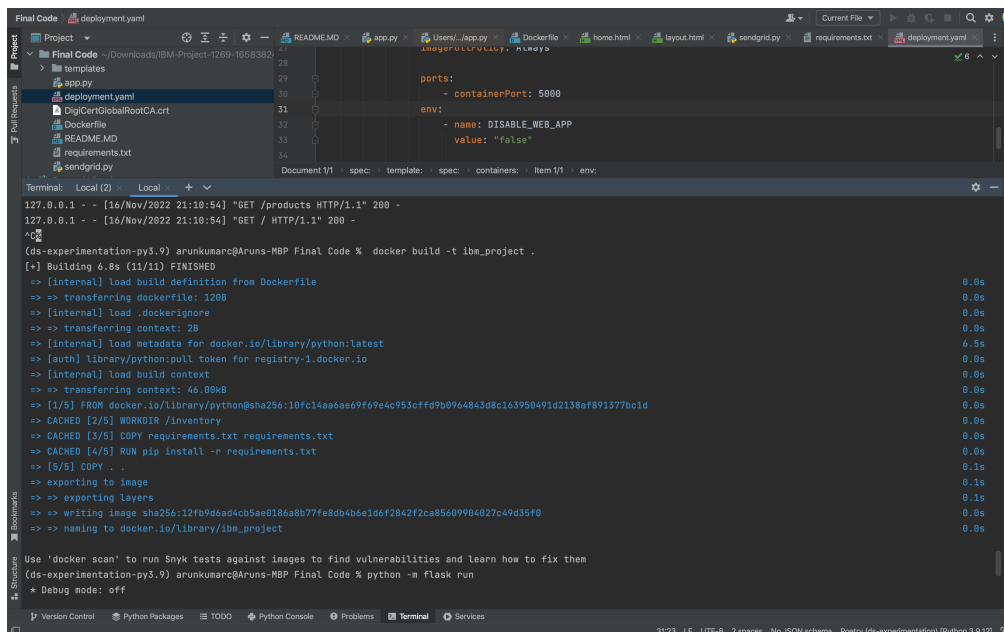
## Sprint 4 (Deploying the application using Docker and Kubernetes):

**Note:** Make sure to create a Dockerfile in the project folder.

Login into DockerHub in Project Folder using command prompt. This connects local docker desktop to cloud docker hub.



Building an image for our project,





OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER

```
arunkumarc@Aruns-MBP Untitled Folder % kubectl apply -f dep.yaml
pod/mypod unchanged
arunkumarc@Aruns-MBP Untitled Folder %
```

The screenshot displays the IBM Cloud Container Registry interface. The left sidebar contains navigation links: IBM Cloud, Container Registry, Quick start, Namespaces (selected), Repositories, Images, Trash, and Settings. The main content area is titled 'Namespaces' and shows a search bar and a 'Location' dropdown set to 'Global'. Below this is a table of namespaces. The table has columns: Name, Resource group, Repository count, Image count, and Retention policy. One namespace is listed: 'cloudprojects' with a 'Default' resource group, 1 repository, 1 image, and a retention policy of 'Retain all images'. The bottom of the table shows 'Items per page: 25' and '1-1 of 1 item'. The top navigation bar includes a search bar, 'Catalog', 'Manage', and the user's account name 'Arun Kumar's Account'.

Name	Resource group	Repository count	Image count	Retention policy
cloudprojects	Default	1	1	Retain all images

```

last login: Wed Nov 16 21:12:15 on tty002
carunkumar@Aruns-MacBook-Pro: ~ % lscloud login
API endpoint: https://cloud.ibm.com

Email: carunkumar19c@erishakthi.ac.in

>Password:
Authentication...
Credentials were rejected.
Code: BXN1M600E, message: The credentials you entered for the user 'carunkumar19c@erishakthi.ac.in' are incorrect.

>Password:
Authentication...
OK

Targeted account 'Arun Kumar's Account [82f35ef9ae576280b8a7cc08f41edcd]

Select a region (or press enter to skip):
1. us-east
2. in-che
3. jp-osa
4. jp-tok
5. kr-seo
6. mumbai
7. eu-gb
8. ca-tor
9. us-south
10. us-west
11. br-sao

Enter a number:

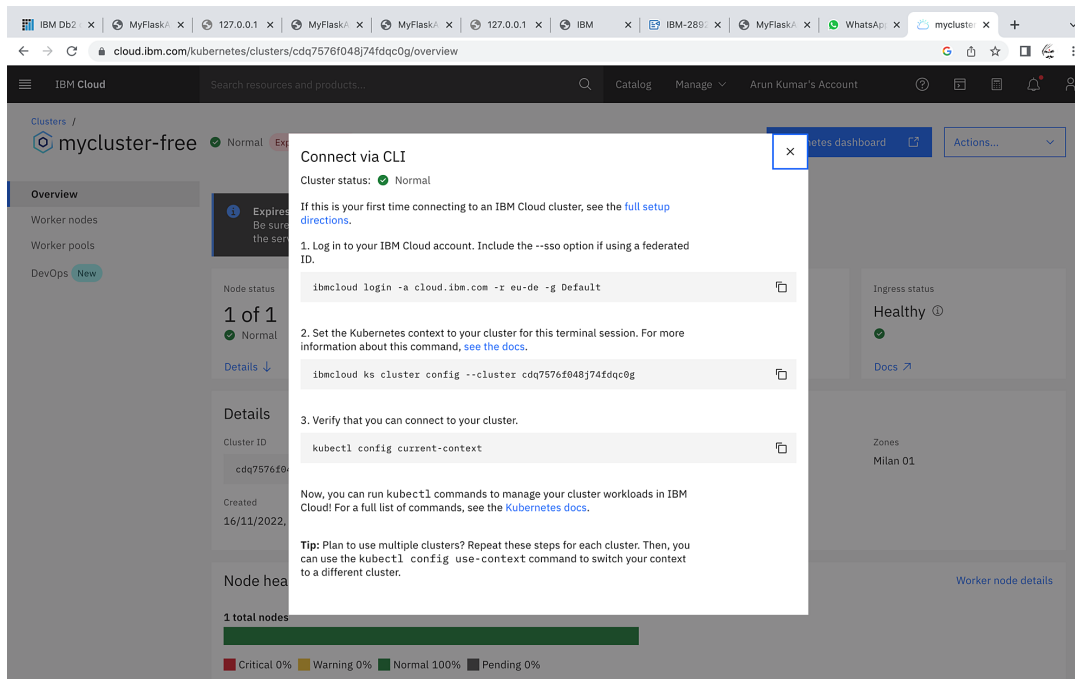
API endpoint: https://cloud.ibm.com
Region:
User: carunkumar19c@erishakthi.ac.in
Account: Arun Kumar's Account [82f35ef9ae576280b8a7cc08f41edcd]
Resource group: No resource group targeted, use 'lscloud target -g RESOURCE_GROUP'
OR API endpoint:
Org:
Space:
carunkumar@Aruns-MacBook-Pro: ~ % lscloud cr login
Logging 'docker' in to 'icr.io'...
logged in to 'icr.io'.

OK
carunkumar@Aruns-MacBook-Pro: ~ % docker push icr.io/cloudprojects/inventory:inventory-22.11.16
The push refers to repository [icr.io/cloudprojects/inventory]
0b235f86d8c4: Layer already exists
8b82d47c6d4d: Pushing [ ] 2.181MB/191.6MB
f80a5a08077f: Layer already exists
6e18a249b3a3: Layer already exists
3d6c7a52b3a3: Pushed
24bf86d8c4e4: Pushed
1b6c7a52b3a3: Pushing [ ] 4.939MB/47.85MB
e6a9b5c4a999: Pushed
397a32a48b30: Pushing [ ] 286.19B/52.9MB
89c324a487b2: Pushing [ ]
802310d3140c: Pushed
f1c122f9b8a3: Pushed
c6ba2969378f: Waiting

```

**Note:** Create a Kubernetes Cluster in IBM Cloud and wait for the work node to get fully deployed.

Then, Login into Kubernetes Cluster using the following commands,



The screenshot shows the IBM Cloud Kubernetes dashboard for a cluster named 'mycluster-free'. A modal window titled 'Connect via CLI' is open, providing instructions for connecting to the cluster via the command line interface (CLI). The modal includes the following steps:

- Log in to your IBM Cloud account. Include the `--sso` option if using a federated ID.  
`ibmcloud login -a cloud.ibm.com -r eu-de -g Default`
- Set the Kubernetes context to your cluster for this terminal session. For more information about this command, [see the docs](#).  
`ibmcloud ks cluster config --cluster cdq7576f048j74fdqc0g`
- Verify that you can connect to your cluster.  
`kubectl config current-context`

Below the steps, a tip states: 'Now, you can run kubectl commands to manage your cluster workloads in IBM Cloud! For a full list of commands, see the [Kubernetes docs](#).  
**Tip:** Plan to use multiple clusters? Repeat these steps for each cluster. Then, you can use the `kubectl config use-context` command to switch your context to a different cluster.'

The background dashboard shows the cluster status as 'Normal' and 'Healthy'. It also displays a 'Node status' section with '1 of 1' nodes listed as 'Normal'. A legend at the bottom indicates the status of the nodes: Critical 0%, Warning 0%, Normal 100%, and Pending 0%.

Check for the public IP address in your IBM Kubernetes Cluster under Worker Node,

IBM Db2 x MyFlask x 127.0.0.1 x MyFlask x MyFlask x 127.0.0.1 x IBM x IBM-28 x MyFlask x WhatsApp x mycluster x

cloud.ibm.com/kubernetes/clusters/cdq7576f048j74fdqc0g/nodes

IBM Cloud Search resources and products... Catalog Manage Arun Kumar's Account

Clusters / mycluster-free Normal Expires in 30 days Add tags Help Kubernetes dashboard Actions...

Overview Worker nodes Worker pools DevOps New

Pool: Filter... Search Add

<input type="checkbox"/>	Name	Status	Worker pool	Zone	Private IP	Public IP	Version
<input checked="" type="checkbox"/>	000000a9	Normal	default	Milan 01	10.144.217.156	159.122.179.109	1.24.7_1543

Items per page: 25 1-1 of 1 item 1 1 of 1 page

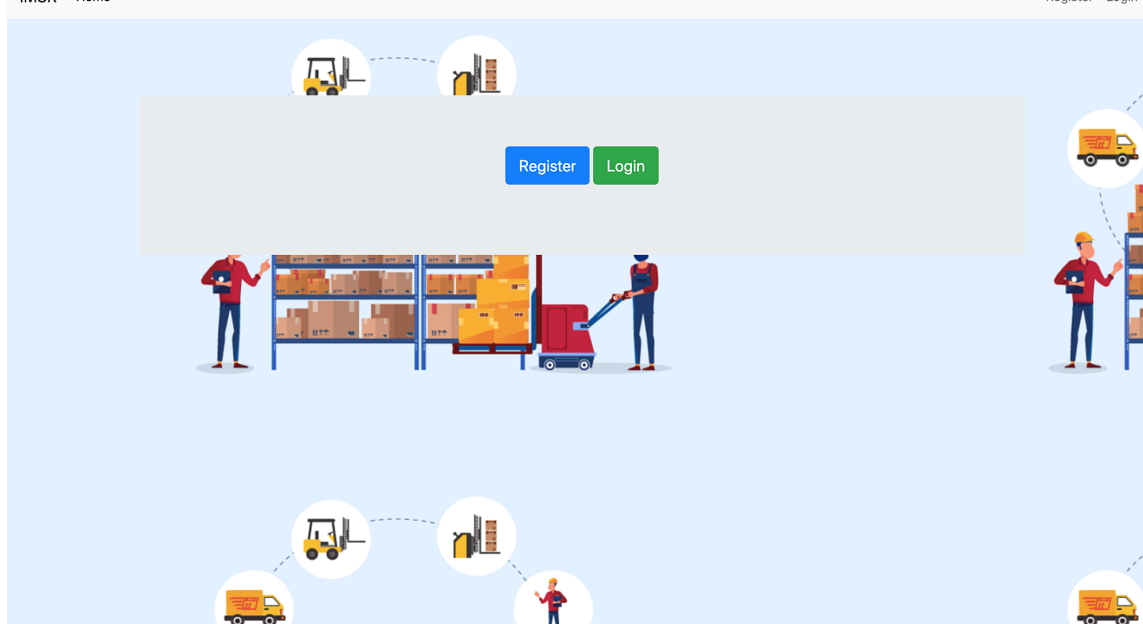
Thus we have the Public IP address and the Nodeport. so can access application

IBM Db2 x MyFlask x 127.0.0.1 x MyFlask x MyFlask x 127.0.0.1 x IBM x IBM-28 x MyFlask x WhatsApp x mycluster x Debug x mycluster x API key x My x Worker x

Not Secure 159.122.179.109:31752

IMSR Home Register Login

Register Login



Our application ip :

<http://159.122.179.109:31752/>

**Result:**

Thus In this way We developed a “Inventory management System for Retailers” using Python, Sendgrid and IBM Cloud Services (IBM DB2, IBM Container registry, IBM Kubernetes).

# Thank You!