Final Deliverables
Report

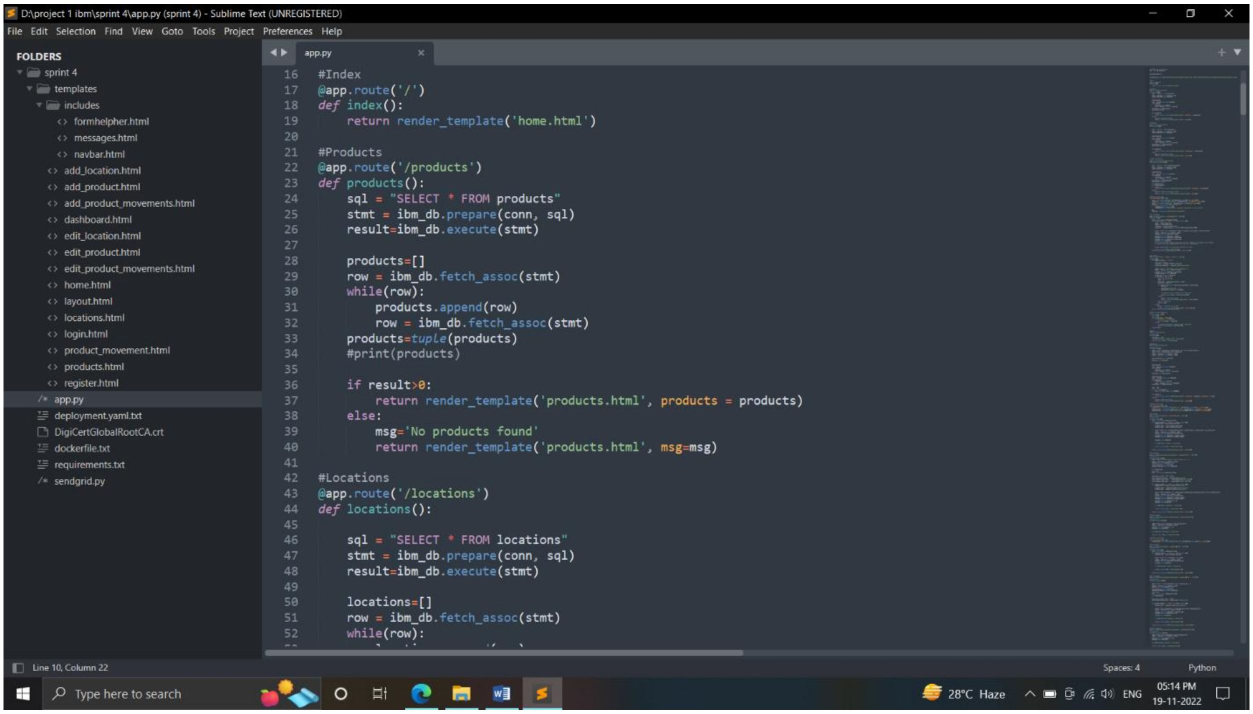
Team ID	PNT2022TMID00278
Project Name	Project – INVENTORY MANAGEMENY

Introduction:

1. Sprint 1 – Backend
2. Sprint 2 – Frontend
3. Sprint 3 – IBM Cloud Integration + Integration of SendGrid
4. Sprint 4 – Deploying the application using Docker and Kubernetes

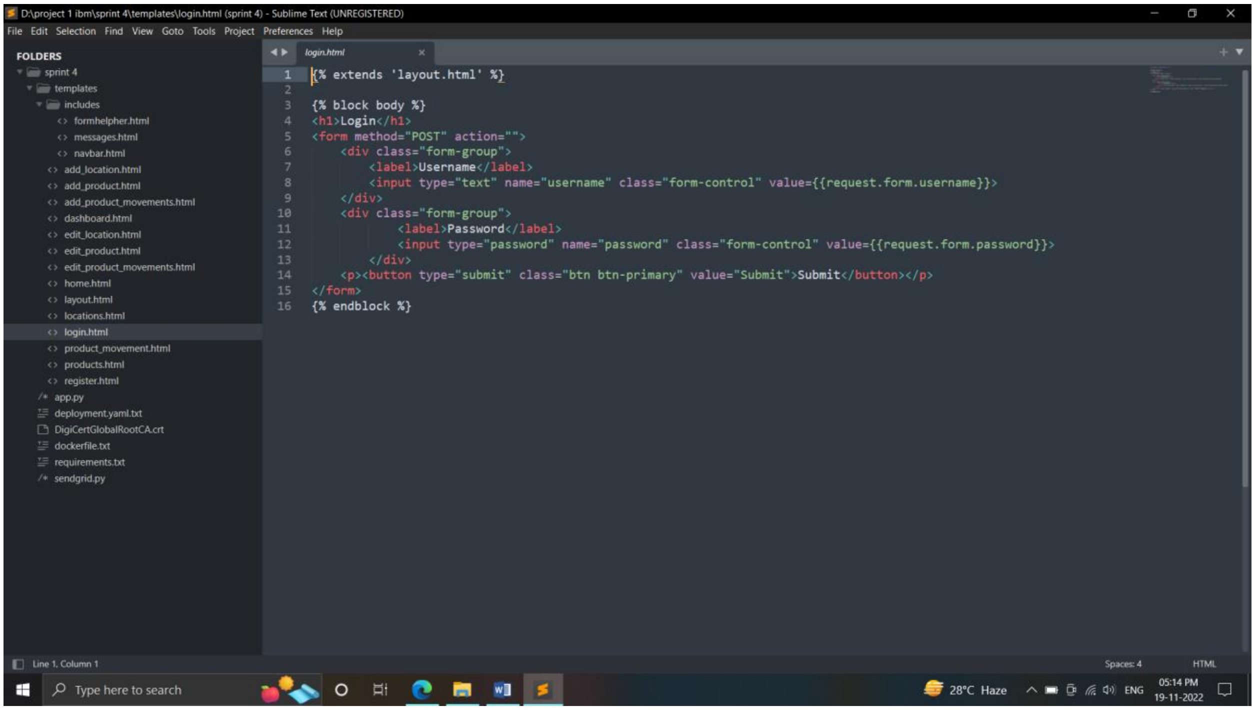
The Team:
Jason Rajkumar A(Team lead)
Abish Kumar P
Abishek Jarvis D
Jaiganesh M

Sprint 1 – Backend:



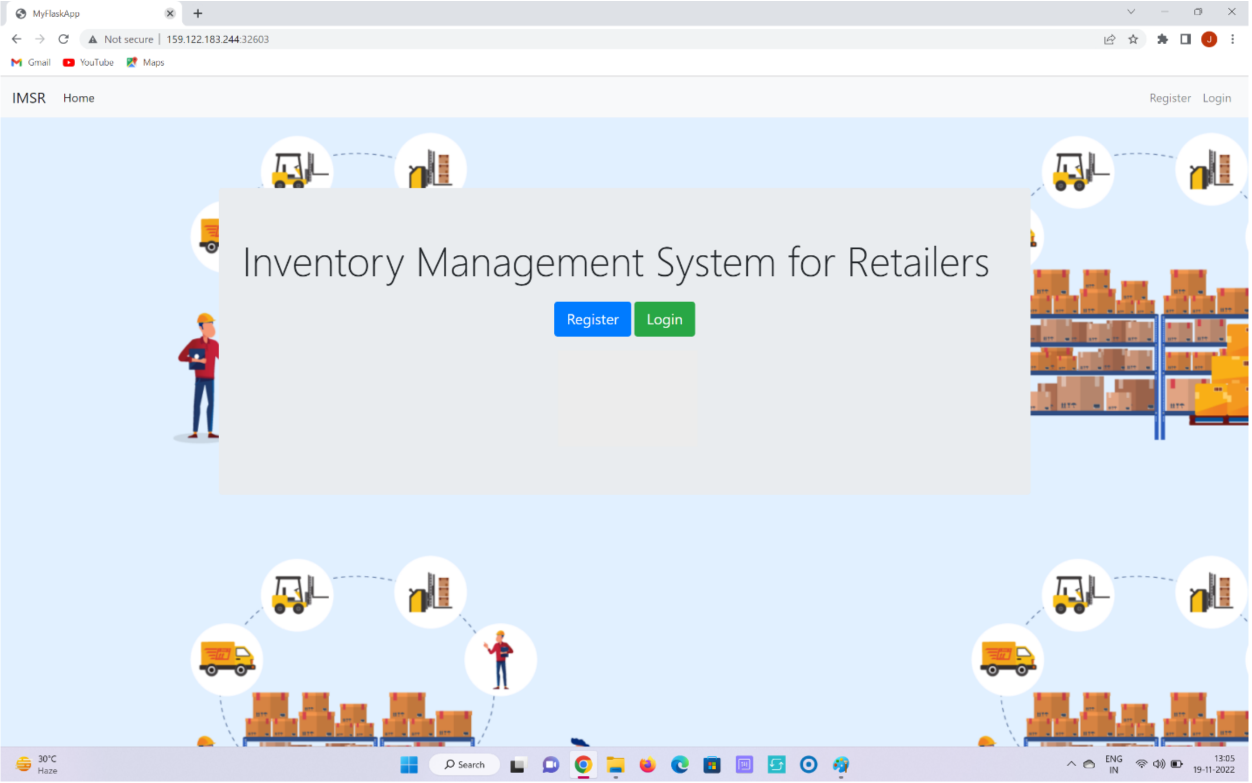
```python
#Index
@app.route('/')
def index():
    return render_template('home.html')

#Products
@app.route('/products')
def products():
    sql = "SELECT * FROM products"
    stmt = ibm_db.prepare(conn, sql)
    result=ibm_db.execute(stmt)

    products=[]
    row = ibm_db.fetch_assoc(stmt)
    while(row):
        products.append(row)
        row = ibm_db.fetch_assoc(stmt)
    products=tuple(products)
    #print(products)

    if result>0:
        return render_template('products.html', products = products)
    else:
        msg='No products found'
        return render_template('products.html', msg=msg)

#Locations
@app.route('/locations')
def locations():

    sql = "SELECT * FROM locations"
    stmt = ibm_db.prepare(conn, sql)
    result=ibm_db.execute(stmt)

    locations=[]
    row = ibm_db.fetch_assoc(stmt)
    while(row):
```

Sprint 2 – Frontend:



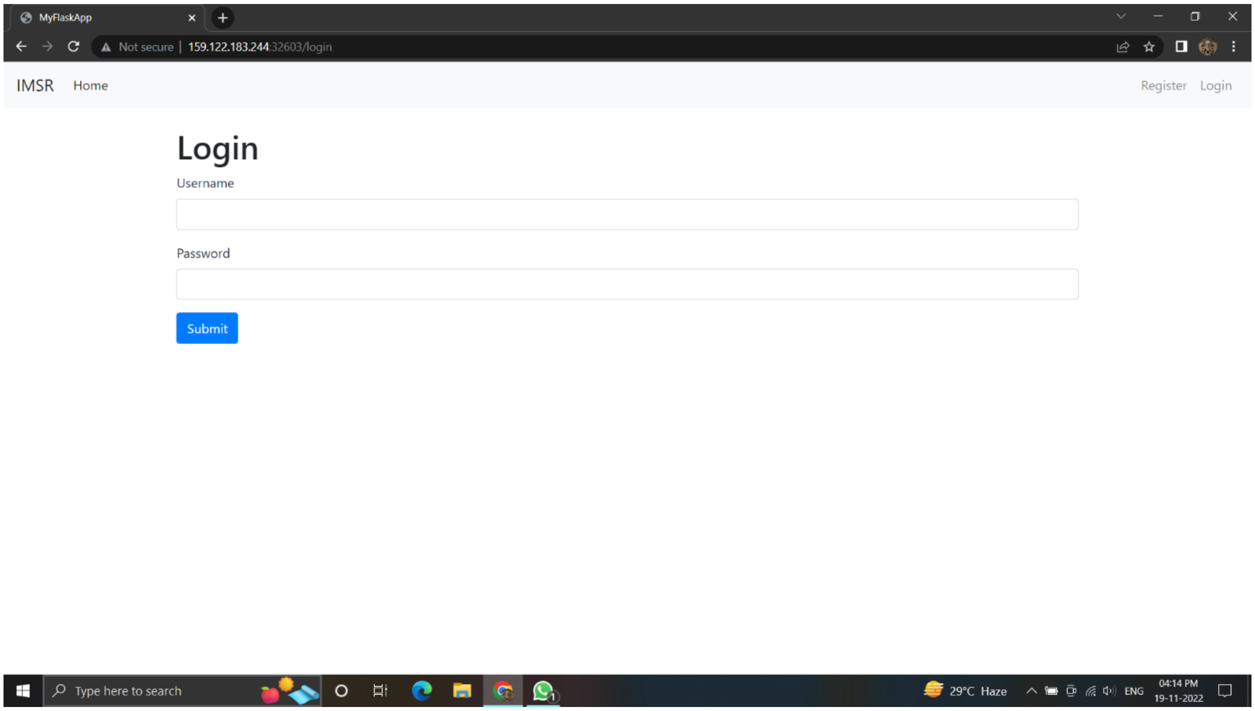
```
{% extends 'layout.html' %}

{% block body %}
<h1>Login</h1>
<form method="POST" action="">
    <div class="form-group">
        <label>Username</label>
        <input type="text" name="username" class="form-control" value={{request.form.username}}>
    </div>
    <div class="form-group">
            <label>Password</label>
            <input type="password" name="password" class="form-control" value={{request.form.password}}>
        </div>
    <p><button type="submit" class="btn btn-primary" value="Submit">Submit</button></p>
</form>
{% endblock %}
```

Home Page

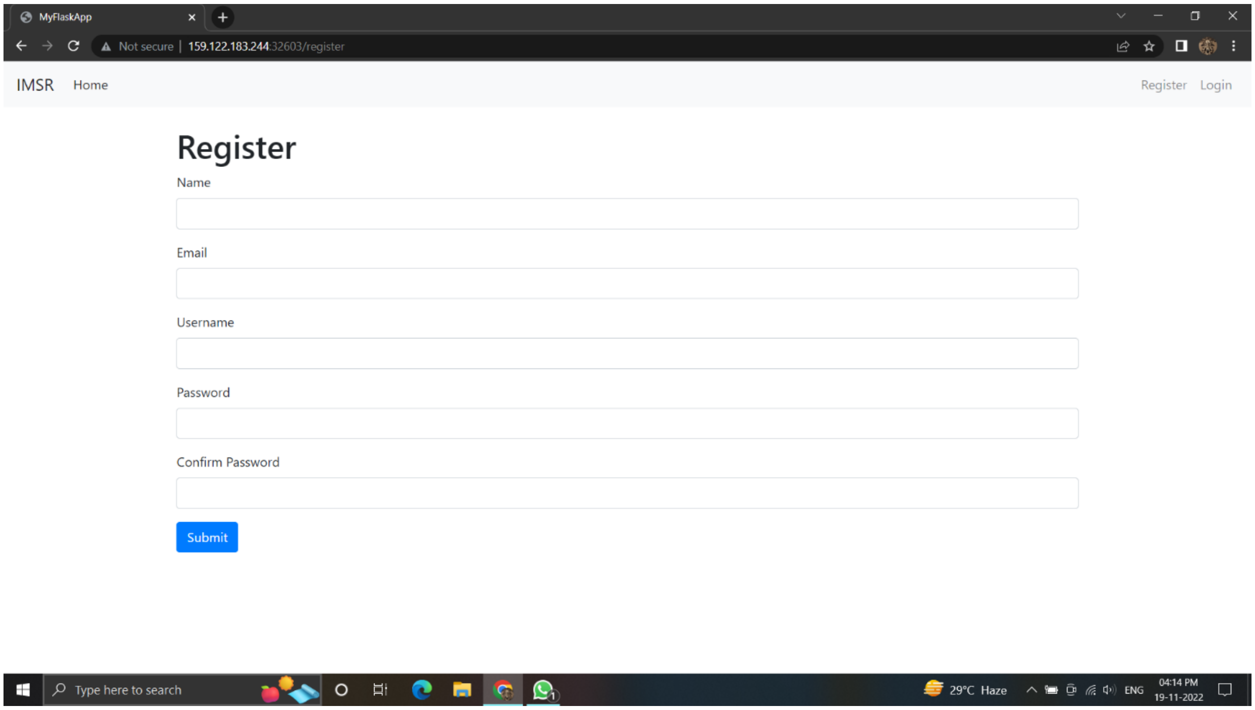


Inventory Management System for Retailers

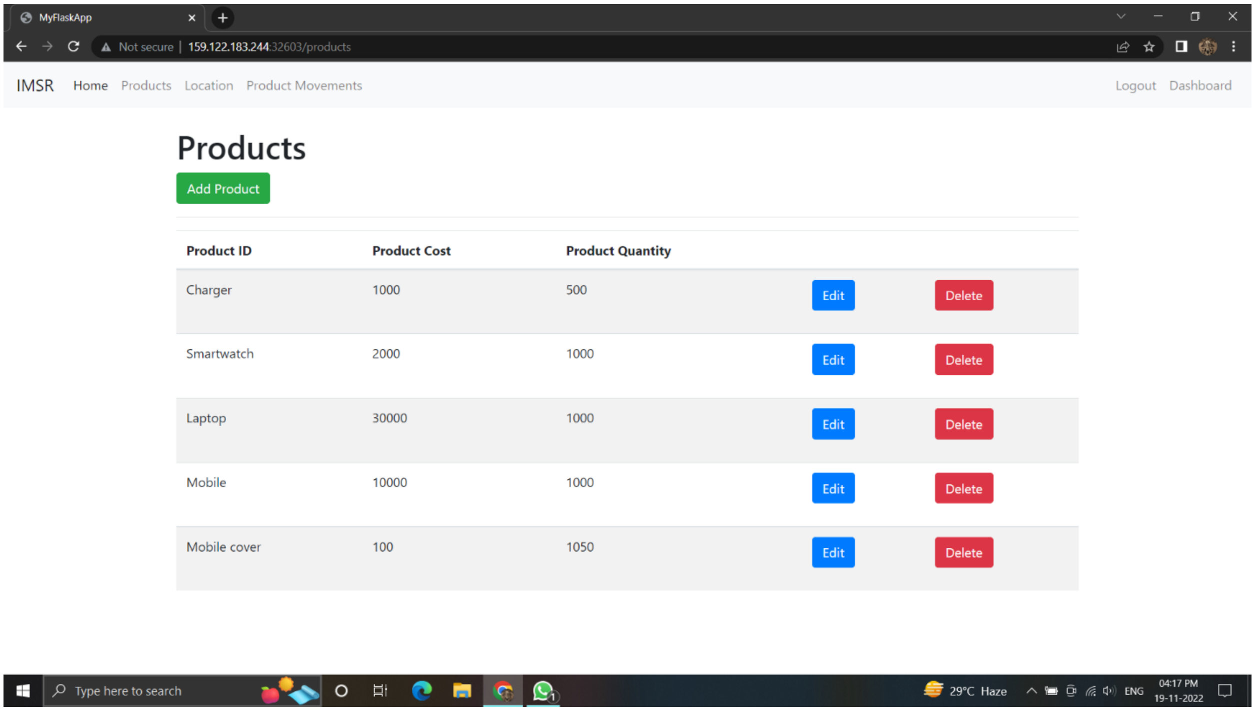
Login


MyFlaskApp
Not secure | 159.122.183.244:32603/login
IMSR
Home
Register
Login
Login
Username
Password
Submit
Type here to search
29°C Haze
ENG
04:14 PM
19-11-2022

Register Page



Products Page

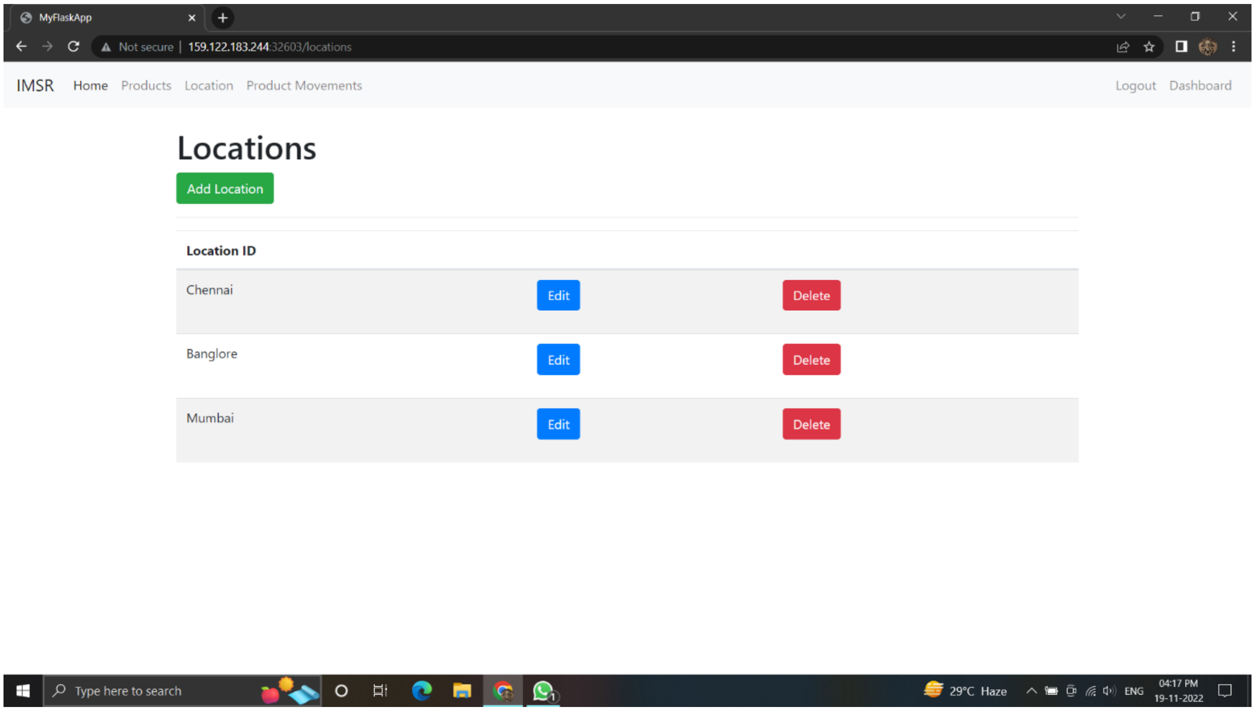


Products

Add Product

Product ID	Product Cost	Product Quantity		
Charger	1000	500	Edit	Delete
Smartwatch	2000	1000	Edit	Delete
Laptop	30000	1000	Edit	Delete
Mobile	10000	1000	Edit	Delete
Mobile cover	100	1050	Edit	Delete

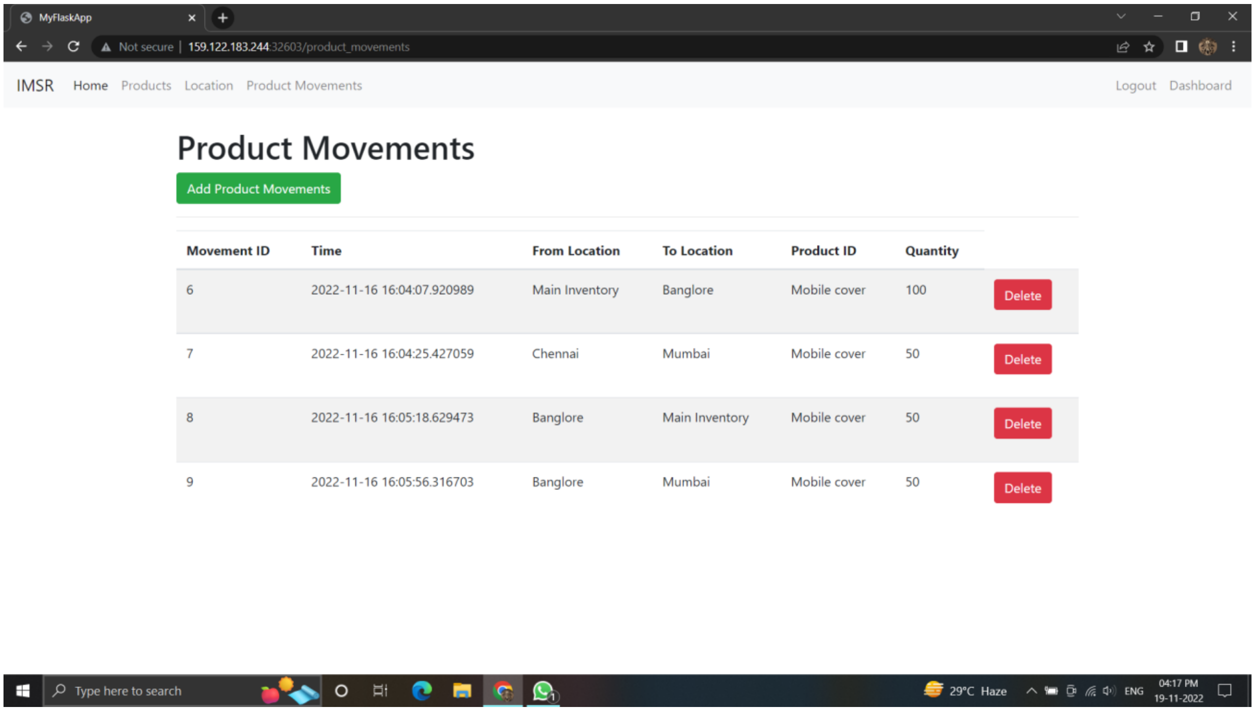
Location



Locations

Location ID		
Chennai	Edit	Delete
Banglore	Edit	Delete
Mumbai	Edit	Delete

Product Movements Page



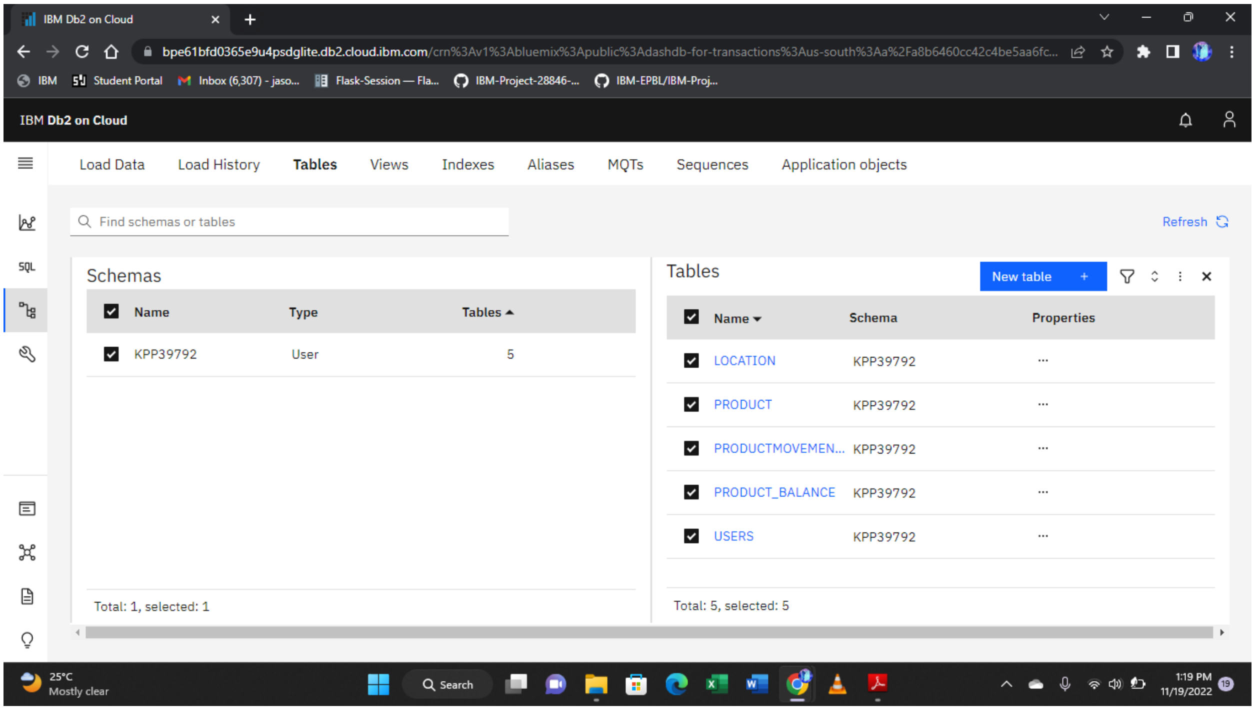
Product Movements

Add Product Movements

Movement ID	Time	From Location	To Location	Product ID	Quantity	
6	2022-11-16 16:04:07.920989	Main Inventory	Banglore	Mobile cover	100	Delete
7	2022-11-16 16:04:25.427059	Chennai	Mumbai	Mobile cover	50	Delete
8	2022-11-16 16:05:18.629473	Banglore	Main Inventory	Mobile cover	50	Delete
9	2022-11-16 16:05:56.316703	Banglore	Mumbai	Mobile cover	50	Delete

Sprint 3 - IBM Cloud Integration + Integration of SendGrid:

IBM CLOUD INTEGRATION



Schemas

☑	Name	Type	Tables ▲
☑	KPP39792	User	5

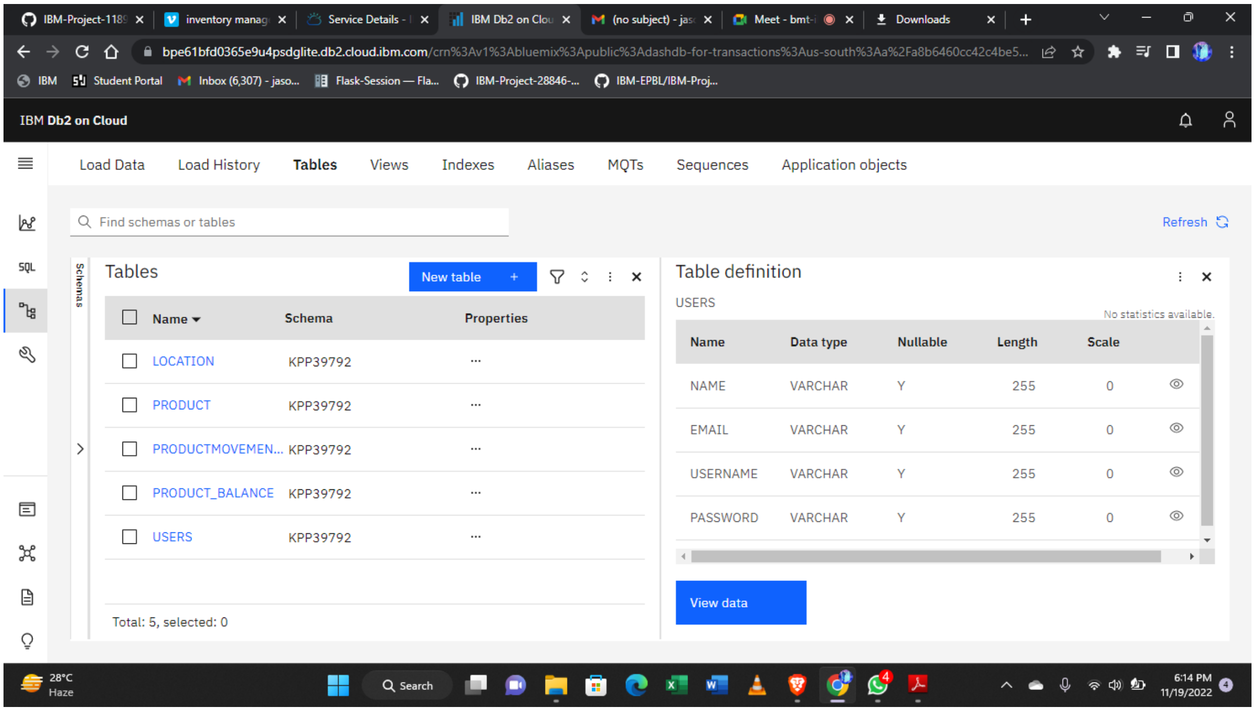
Total: 1, selected: 1

Tables

☑	Name ▾	Schema	Properties
☑	LOCATION	KPP39792	···
☑	PRODUCT	KPP39792	···
☑	PRODUCTMOVEMEN...	KPP39792	···
☑	PRODUCT_BALANCE	KPP39792	···
☑	USERS	KPP39792	···

Total: 5, selected: 5

Schema of the particular table (For Example, Product_Balance)



Tables

□	Name	Schema	Properties
□	LOCATION	KPP39792	···
□	PRODUCT	KPP39792	···
□	PRODUCTMOVEMEN...	KPP39792	···
□	PRODUCT_BALANCE	KPP39792	···
□	USERS	KPP39792	···

Total: 5, selected: 0

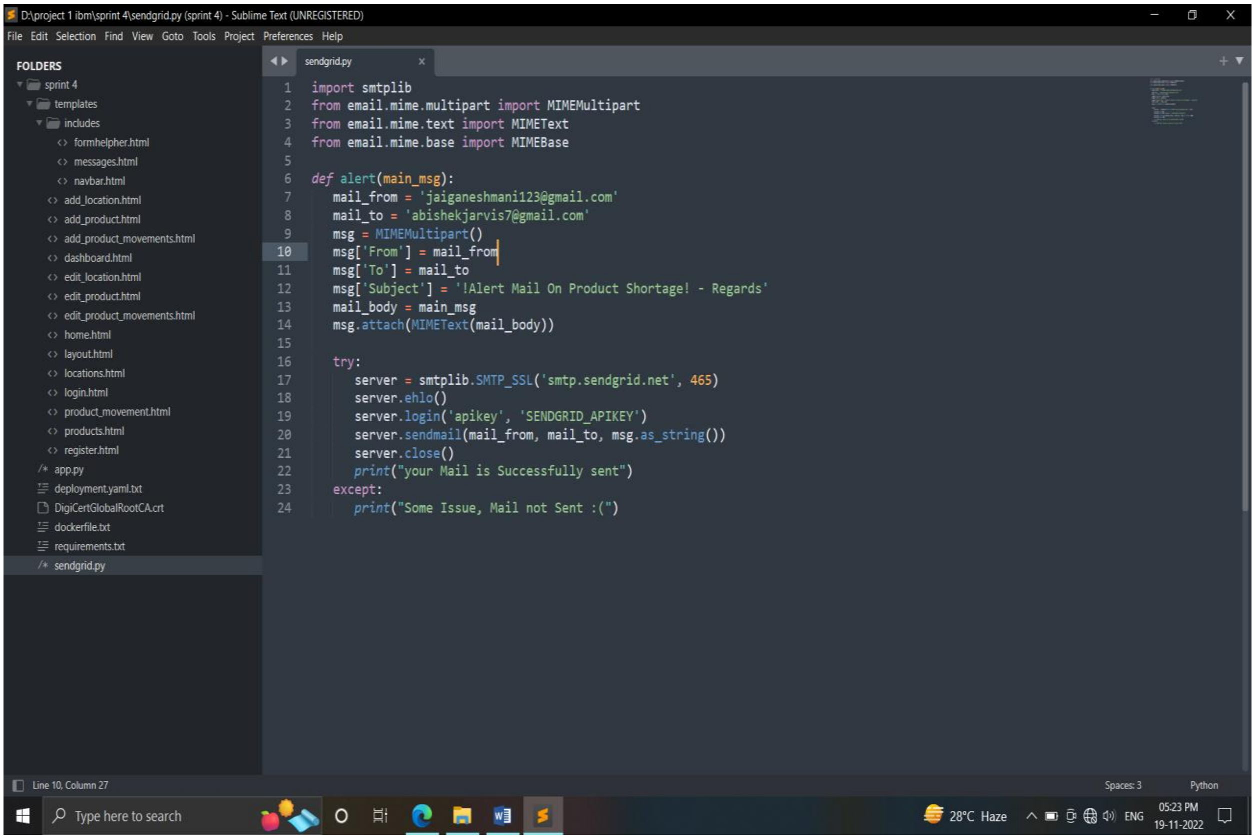
Table definition

USERS

No statistics available.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
EMAIL	VARCHAR	Y	255	0
USERNAME	VARCHAR	Y	255	0
PASSWORD	VARCHAR	Y	255	0

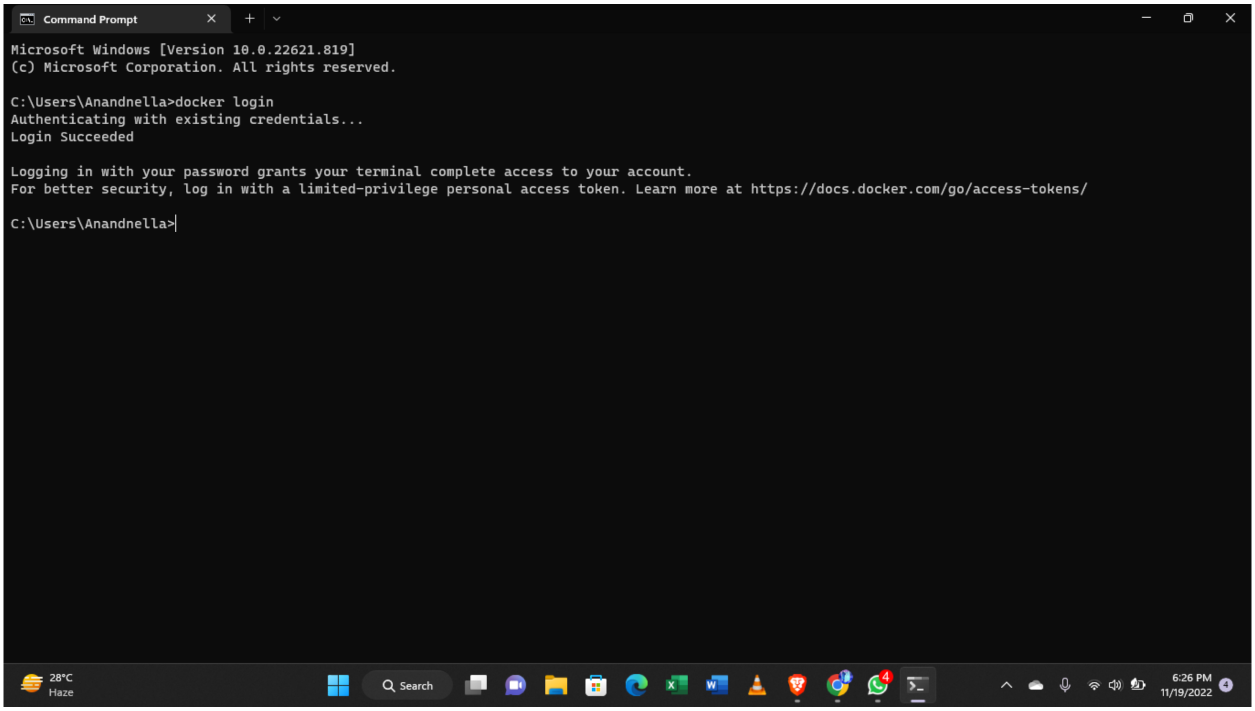
Code for Email alert



```python
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase

def alert(main_msg):
   mail_from = 'jaiganeshmani123@gmail.com'
   mail_to = 'abishekjarvis7@gmail.com'
   msg = MIMEMultipart()
   msg['From'] = mail_from
   msg['To'] = mail_to
   msg['Subject'] = '!Alert Mail On Product Shortage! - Regards'
   mail_body = main_msg
   msg.attach(MIMEText(mail_body))

   try:
      server = smtplib.SMTP_SSL('smtp.sendgrid.net', 465)
      server.ehlo()
      server.login('apikey', 'SENDGRID_APIKEY')
      server.sendmail(mail_from, mail_to, msg.as_string())
      server.close()
      print("your Mail is Successfully sent")
   except:
      print("Some Issue, Mail not Sent :(")
```

Sprint 4 Deploying the application using Docker and Kubernetes

1. **Login into DockerHub in Project Folder using command prompt. This connects local docker desktop to cloud docker hub.**
2. **Build an image for the project.**

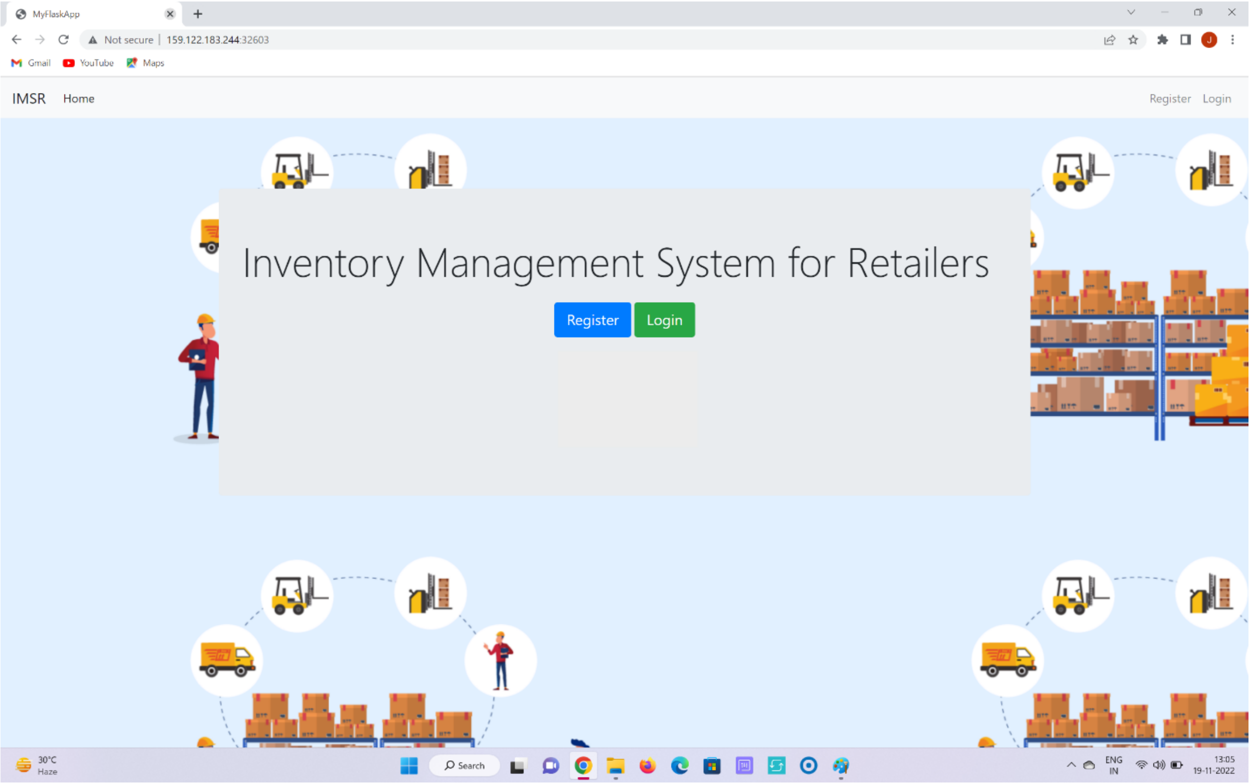


```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anandnella>docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

C:\Users\Anandnella>
```

- **Create a valid Deployment.yaml file.**
- **Create a namespace in IBM Container registry.**

- **Pushing the project into IBM container Registry.**
- **Create a Kubernetes Cluster in IBM Cloud and wait for the work node to get fully deployed.**
- **Check for the public IP address in your IBM Kubernetes Cluster under Worker Node.**

Now, check the public IP address to view your website.



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).