

Final Deliverables Report

Team ID	PNT2022TMID00278
Project Name	Project – INVENTORY MANAGEMENT

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:

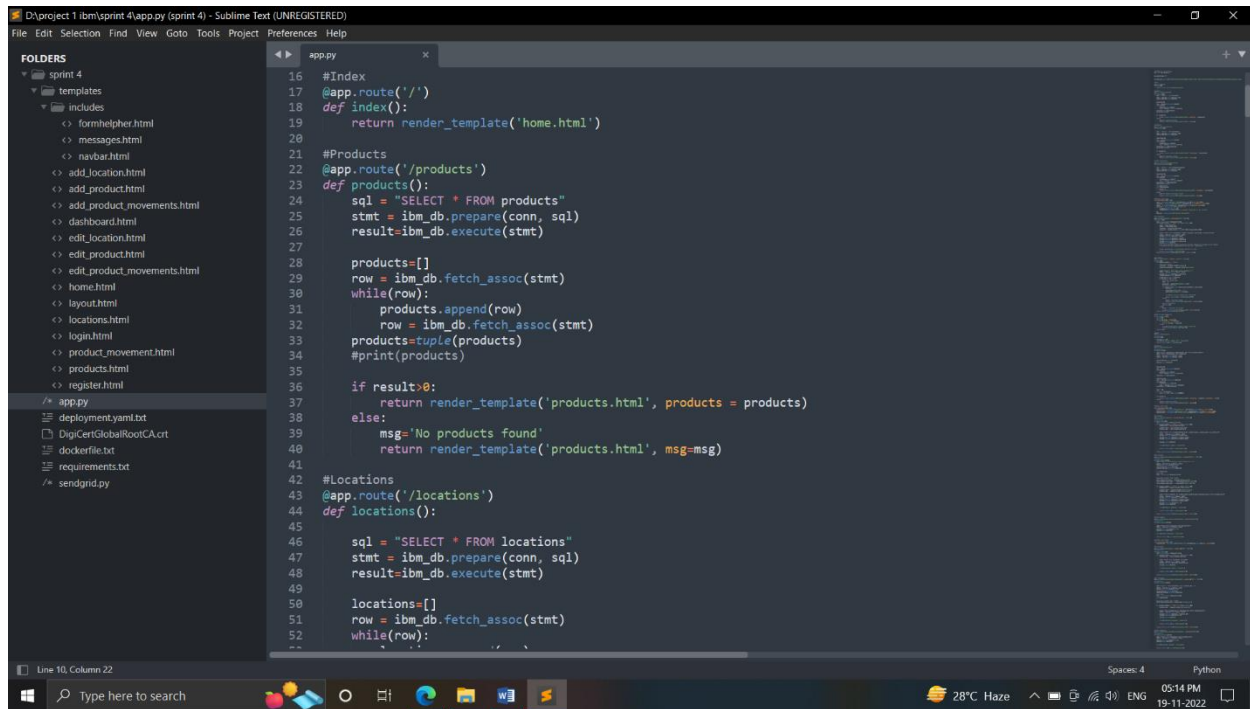
Jason Rajkumar A(Team lead)

Abish Kumar P

Abishek Jarvis D

Jaiganesh M

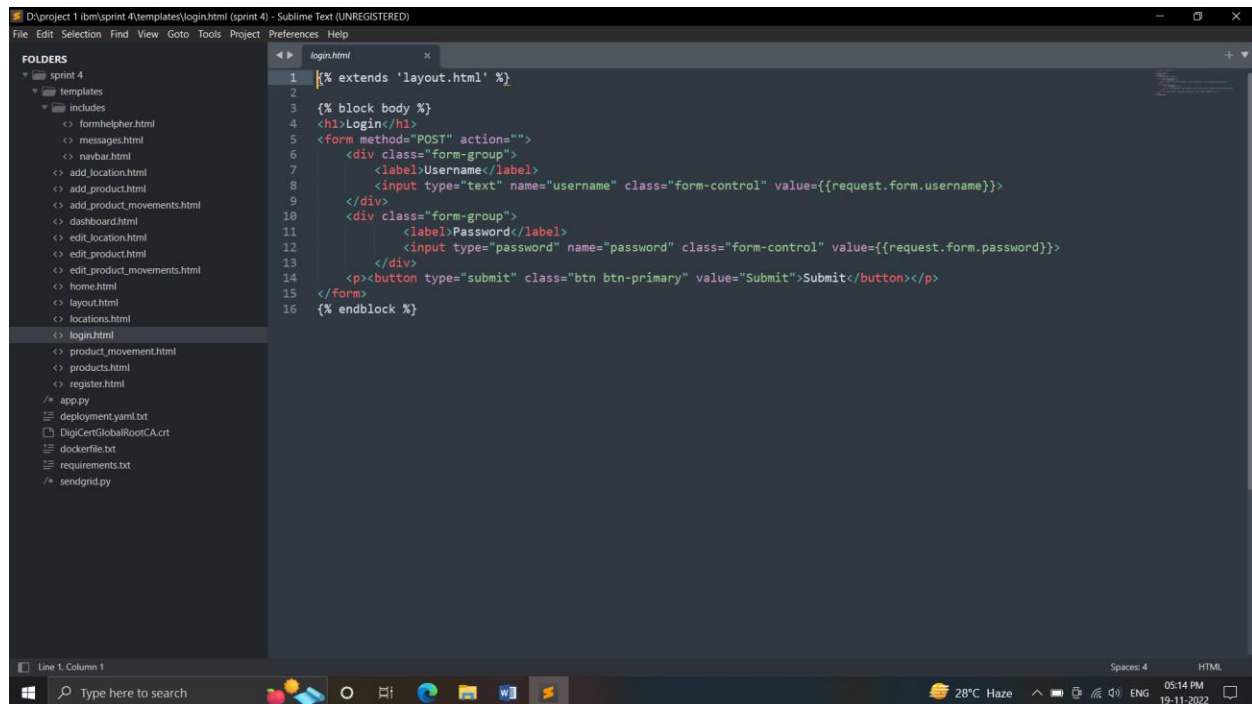
Sprint 1 – Backend:



```
16 #Index
17 @app.route('/')
18 def index():
19     return render_template('home.html')
20
21 #Products
22 @app.route('/products')
23 def products():
24     sql = "SELECT * FROM products"
25     stmt = ibm_db.prepare(conn, sql)
26     result=ibm_db.execute(stmt)
27
28     products=[]
29     row = ibm_db.fetch_assoc(stmt)
30     while(row):
31         products.append(row)
32         row = ibm_db.fetch_assoc(stmt)
33     products=tuple(products)
34     #print(products)
35
36     if result>0:
37         return render_template('products.html', products = products)
38     else:
39         msg='No products found'
40         return render_template('products.html', msg=msg)
41
42 #Locations
43 @app.route('/locations')
44 def locations():
45
46     sql = "SELECT * FROM locations"
47     stmt = ibm_db.prepare(conn, sql)
48     result=ibm_db.execute(stmt)
49
50     locations=[]
51     row = ibm_db.fetch_assoc(stmt)
52     while(row):
53         locations.append(row)
54         row = ibm_db.fetch_assoc(stmt)
55     locations=tuple(locations)
56     #print(locations)
57
58     if result>0:
59         return render_template('locations.html', locations = locations)
60     else:
61         msg='No locations found'
62         return render_template('locations.html', msg=msg)
63
64 #Locations
65 @app.route('/locations')
```

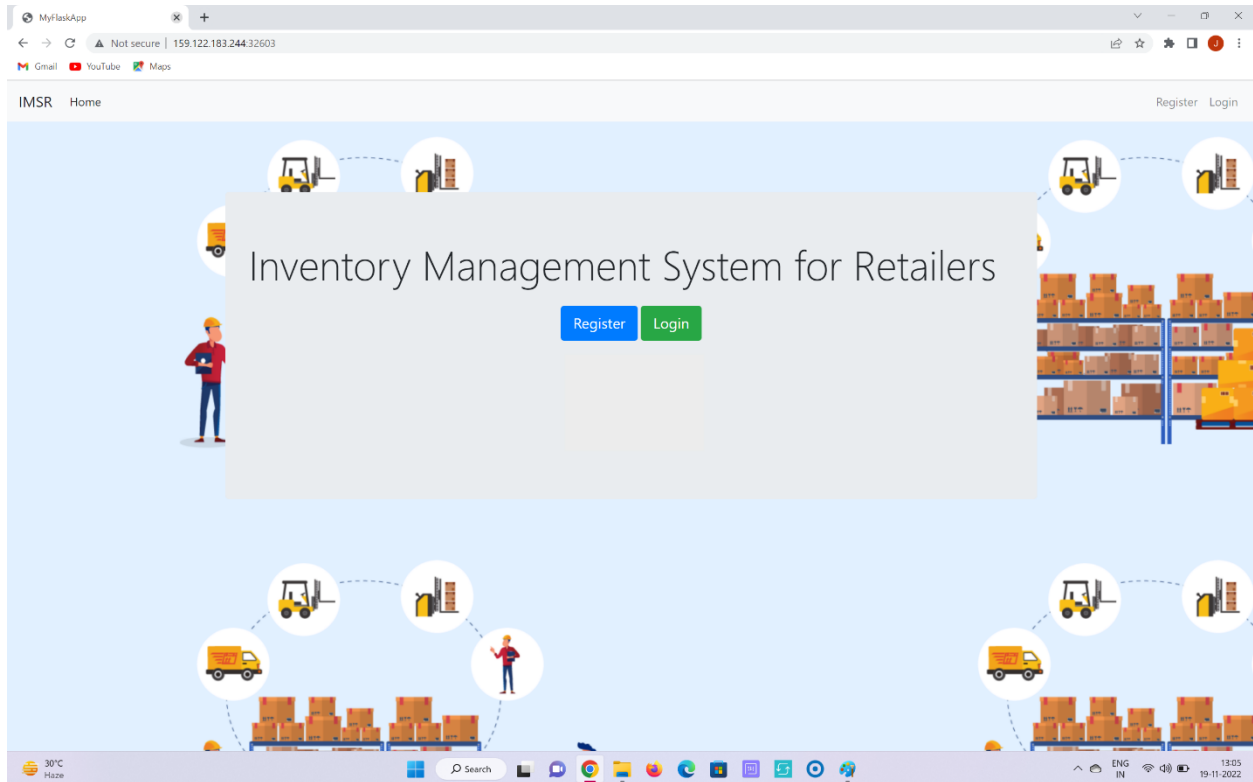
Line 10, Column 22 Spaces: 4 Python

Sprint 2 – Frontend:

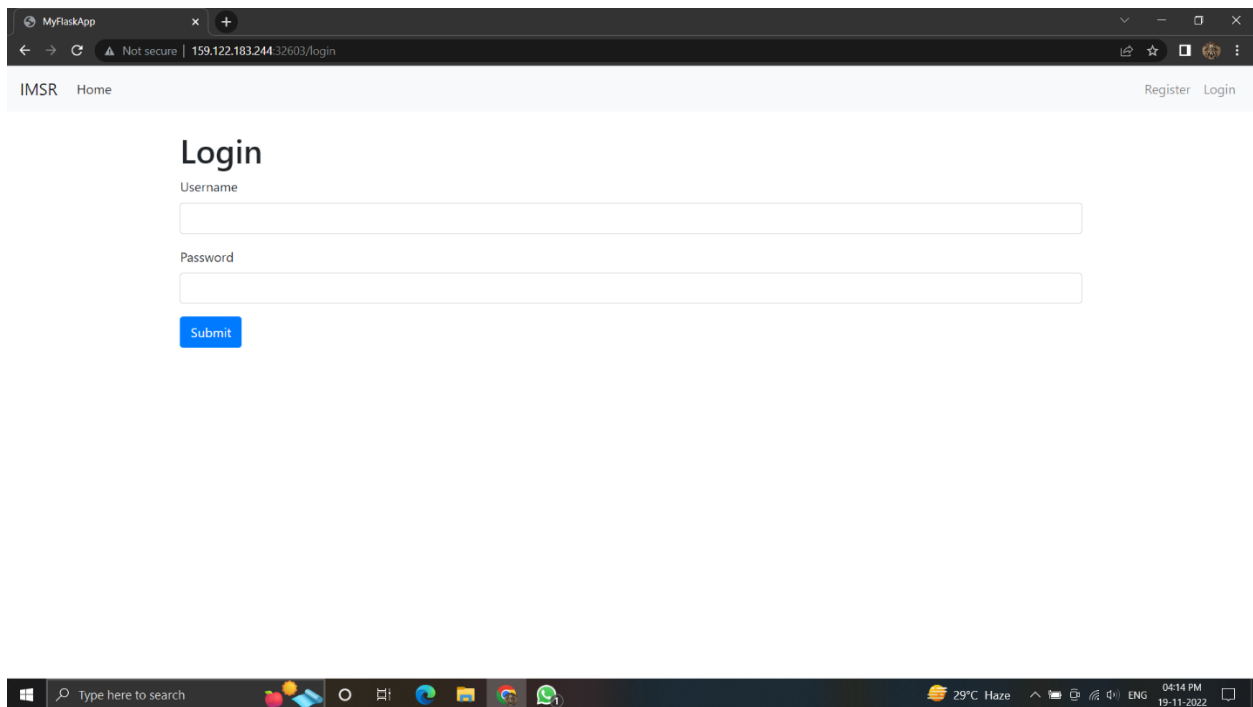


```
1 {% extends 'layout.html' %}
2
3 {% block body %}
4 <h1>Login</h1>
5 <form method="POST" action="">
6   <div class="form-group">
7     <label>Username</label>
8     <input type="text" name="username" class="form-control" value="{{request.form.username}}>
9   </div>
10   <div class="form-group">
11     <label>Password</label>
12     <input type="password" name="password" class="form-control" value="{{request.form.password}}>
13   </div>
14   <p><button type="submit" class="btn btn-primary" value="Submit">Submit</button></p>
15 </form>
16 {% endblock %}
```

Home Page



Login



The screenshot shows a web browser window with the title 'MyFlaskApp'. The address bar indicates the URL is '159.122.183.244:32603/login' and the connection is 'Not secure'. The page has a light gray 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. The Windows taskbar at the bottom shows the search bar, task view, and several application icons including Edge, File Explorer, and WhatsApp. The system tray on the right displays the weather as '29°C Haze' and the date/time as '04:14 PM 19-11-2022'.

MyFlaskApp

Not secure | 159.122.183.244:32603/login

IMSR Home Register Login

Login

Username

Password

Submit

Type here to search

29°C Haze 04:14 PM 19-11-2022

Register Page

MyFlaskApp x +

← → ↻ ⚠ Not secure | 159.122.183.244:32603/register

IMSR Home Register Login

Register

Name

Email

Username

Password

Confirm Password

Submit

Type here to search

29°C Haze 04:14 PM 19-11-2022

Products Page

MyFlaskApp

Not secure | 159.122.183.244:32603/products

LogoutDashboard

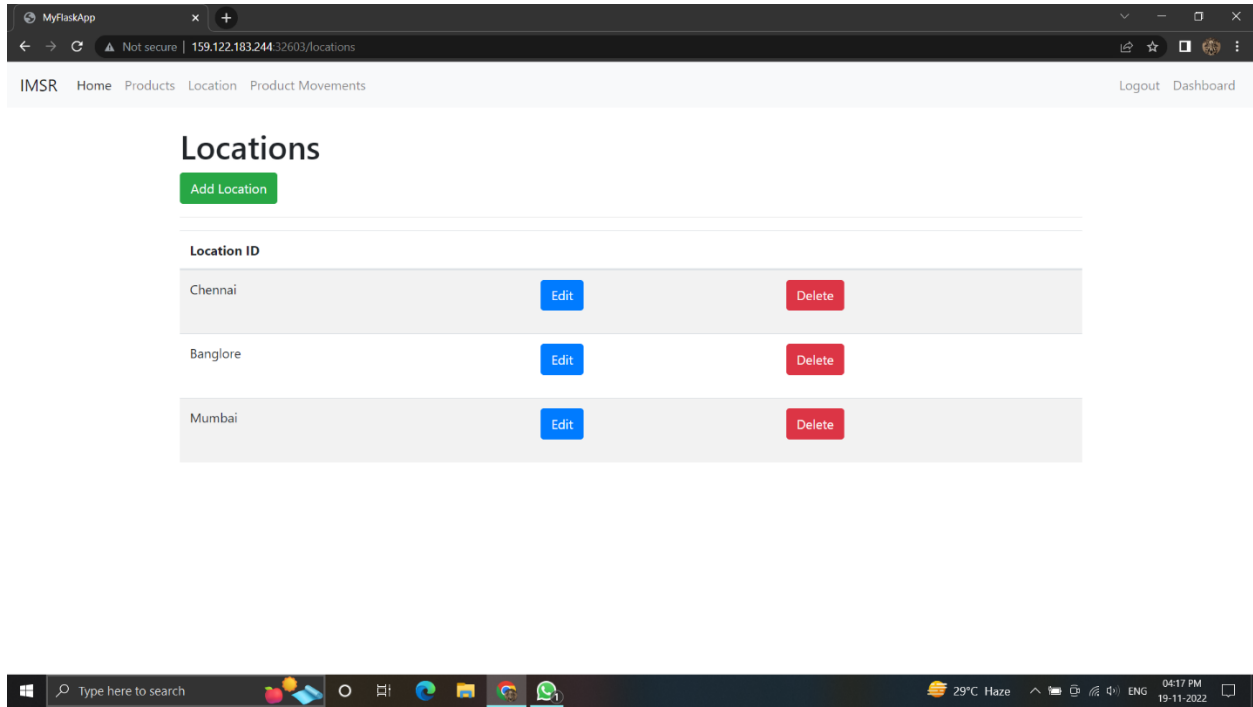
IMSRHomeProductsLocationProduct Movements

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



MyFlaskApp

Not secure | 159.122.183.244:32603/product_movements

IMSRHomeProductsLocationProduct MovementsLogoutDashboard

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

The screenshot displays 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 'Tables' tab is currently selected. Below the navigation bar, there is a search bar labeled 'Find schemas or tables' and a 'Refresh' button. The main content area is divided into two panels: 'Schemas' on the left and 'Tables' on the right. The 'Schemas' panel shows a table with one entry: 'KPP39792' of type 'User'. The 'Tables' panel shows a table with five entries: 'LOCATION', 'PRODUCT', 'PRODUCTMOVEMEN...', 'PRODUCT_BALANCE', and 'USERS', all belonging to the 'KPP39792' schema. A 'New table' button is visible in the top right of the 'Tables' panel. The bottom of the image shows a Windows taskbar with various application icons and a system tray indicating the date and time as 11:19 PM on 11/19/2022.

IBM Db2 on Cloud

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

Find schemas or tables Refresh

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

25°C Mostly clear Search 11:19 PM 11/19/2022

Schema of the particular table (For Example, Product_Balance)

The screenshot displays the IBM Db2 on Cloud console interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is active, showing a list of tables in the KPP39792 schema. The 'PRODUCT_BALANCE' table is selected. The right-hand pane shows the 'Table definition' for 'USERS', which includes columns: NAME (VARCHAR, 255, nullable), EMAIL (VARCHAR, 255, nullable), USERNAME (VARCHAR, 255, nullable), and PASSWORD (VARCHAR, 255, nullable). A 'View data' button is visible at the bottom of the table definition pane. The bottom status bar shows the system temperature (28°C), time (6:14 PM), and date (11/19/2022).

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Tables New table

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

Total: 5, selected: 0

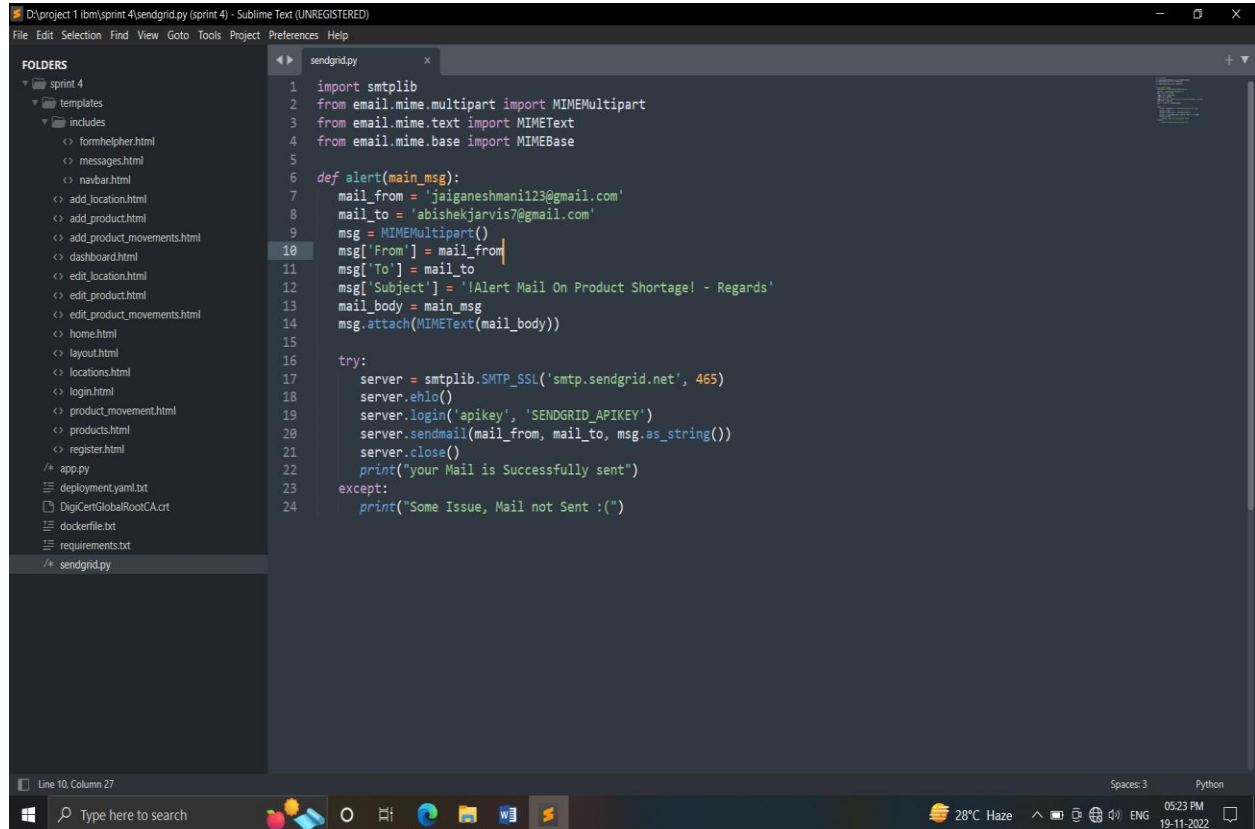
Table definition USERS

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

View data

28°C Haze 6:14 PM 11/19/2022

Code for Email alert



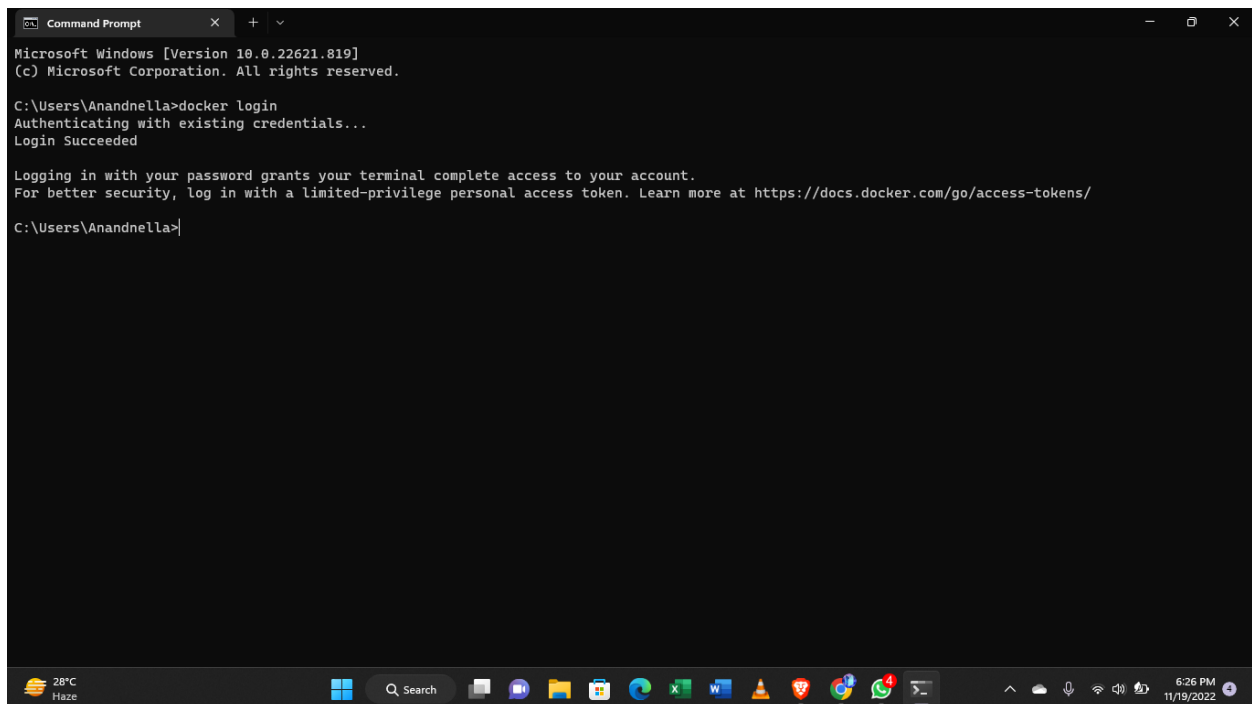
The screenshot shows a Sublime Text editor window titled "D:\project 1 ibm\sprint 4\sendgrid.py (sprint 4) - Sublime Text (UNREGISTERED)". The left sidebar displays a file explorer with a project structure including folders like "templates" and "includes", and various HTML files. The main editor area shows a Python script named "sendgrid.py" with the following code:

```
1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4 from email.mime.base import MIMEBase
5
6 def alert(main_msg):
7     mail_from = 'jaiganeshmani123@gmail.com'
8     mail_to = 'abishekjarvis7@gmail.com'
9     msg = MIMEMultipart()
10    msg['From'] = mail_from
11    msg['To'] = mail_to
12    msg['Subject'] = '!Alert Mail On Product Shortage! - Regards'
13    mail_body = main_msg
14    msg.attach(MIMEText(mail_body))
15
16    try:
17        server = smtplib.SMTP_SSL('smtp.sendgrid.net', 465)
18        server.ehlo()
19        server.login('apikey', 'SENDGRID_APIKEY')
20        server.sendmail(mail_from, mail_to, msg.as_string())
21        server.close()
22        print("your Mail is Successfully sent")
23    except:
24        print("Some Issue, Mail not Sent :(")
```

The status bar at the bottom indicates "Line 10, Column 27", "Spaces: 3", and "Python". The Windows taskbar at the very bottom shows the system clock as 05:23 PM on 19-11-2022, along with weather information (28°C Haze) and various system icons.

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.**



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

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

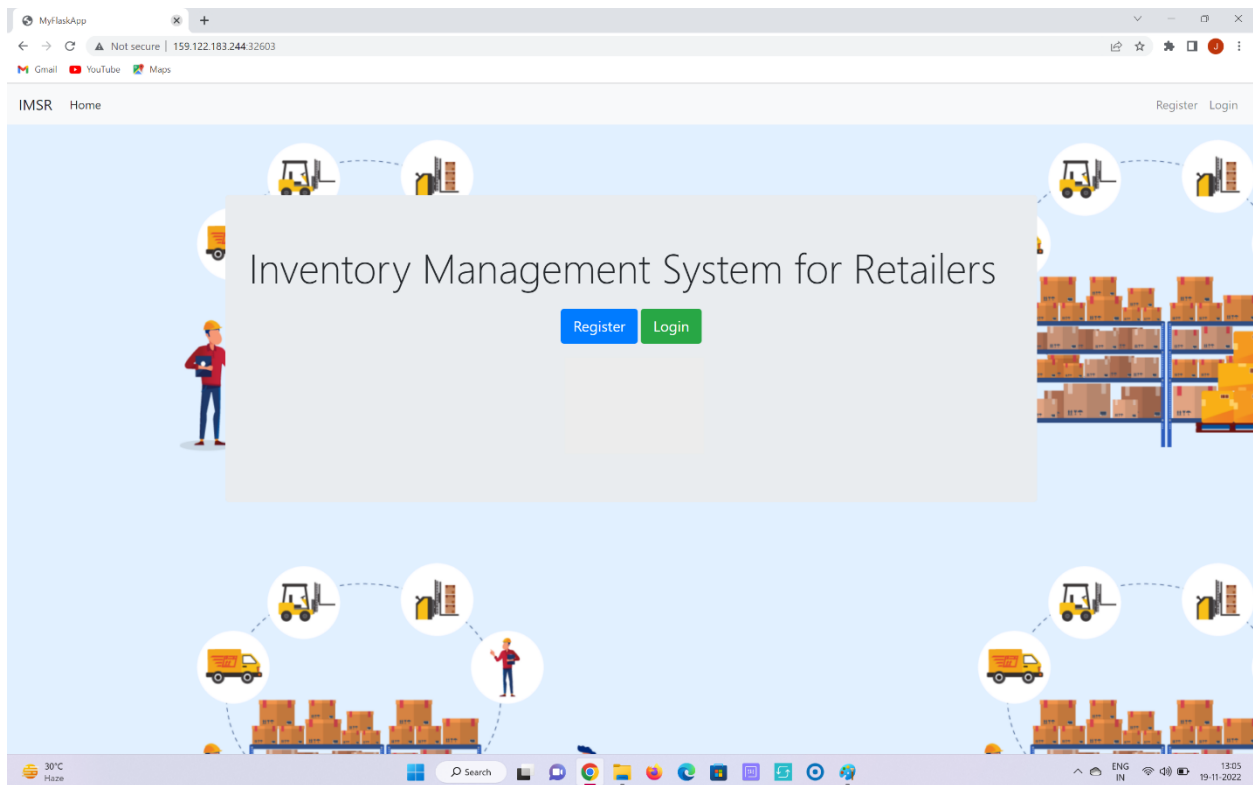
Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

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).