

Date	16 November 2022
Team ID	PNT2022TMID53036
Project Name	Project - Inventory Management System For Retailers

Sprint 3

Pushing the Application into Docker :

```

C:\Windows\System32\cmd.exe

C:\Users\SSN\Desktop\SSN College\IBM\Project Development Phase\Sprint 3>docker build -t sprint3 .
[+] Building 30.2s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.2s
=> => transferring dockerfile: 32B                                              0.0s
=> [internal] load .dockerignore                                                 0.2s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.10.4                29.0s
=> [auth] library/python:pull token for registry-1.docker.io                  0.0s
=> [1/5] FROM docker.io/library/python:3.10.4@sha256:cddbe04ec7846e28870cf8624b46313a22e6407b51ced3776588784caa 0.0s
=> [internal] load build context                                                0.5s
=> => transferring context: 419.84kB                                           0.5s
=> CACHED [2/5] WORKDIR /app                                                    0.0s
=> CACHED [3/5] COPY requirements.txt ./                                        0.0s
=> CACHED [4/5] RUN pip install -r requirements.txt                            0.0s
=> [5/5] COPY . .                                                              0.2s
=> exporting to image                                                           0.1s
=> => exporting layers                                                         0.1s
=> => writing image sha256:8406f1bfd28ad665fb5a3b618451b2dd8a6bec4f6ad10d1b1ec9d45ca7b957cd 0.0s
=> => naming to docker.io/library/sprint3                                     0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Users\SSN\Desktop\SSN College\IBM\Project Development Phase\Sprint 3>

```

Running the docker Container and Docker:

Docker Desktop Upgrade plan

Containers Get feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Only show running containers

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	buoy_khayyam 39b9f600a6c4	sprint2:latest	Exited	5000:5000		
<input type="checkbox"/>	cool_mestorf ac87e8942062	sprint3:latest	Running	5000:5000	31 seconds ago	

Extensions Add Extensions

RAM 1.87GB CPU 0.43% Connected to Hub

Showing 2 items v4.13.1

```

C:\Windows\System32\cmd.exe: docker run -p 5000:5000 sprint3

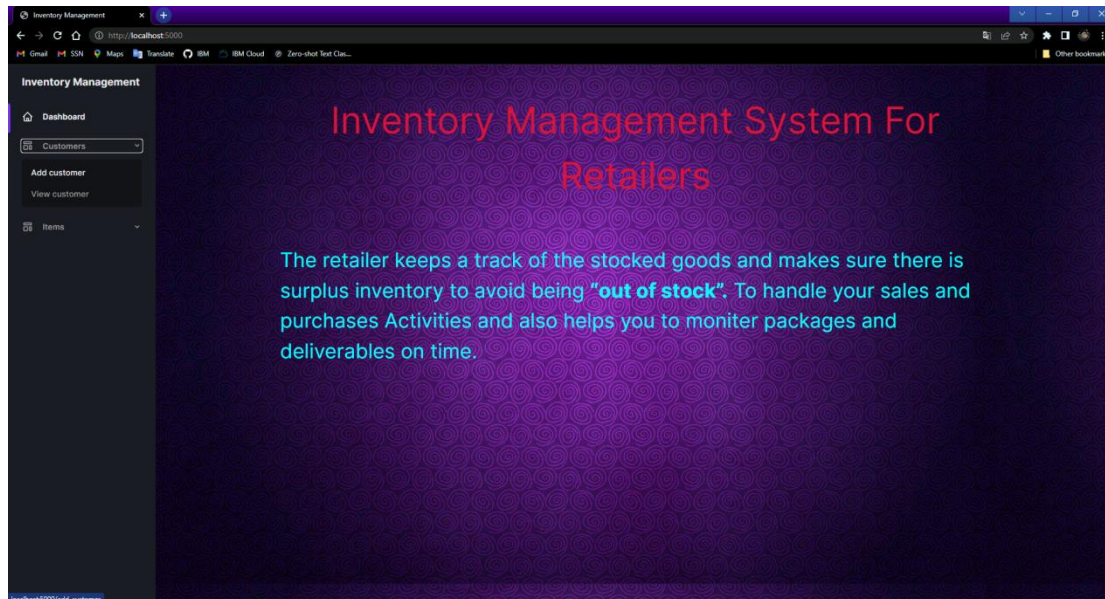
=> [internal] load metadata for docker.io/library/python:3.10.4
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/python:3.10.4@sha256:cddbe04ec7846e28870cf8624b46313a22e6407b51ced3776588784caa
=> [internal] load build context
=> => transferring context: 336.17kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt ./
=> CACHED [4/5] RUN pip install -r requirements.txt
=> [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:1886daef80a7757c06f93e5db3080bf2f5a3732ba0530ba05311a10729a5dc
=> => naming to docker.io/library/sprint3

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

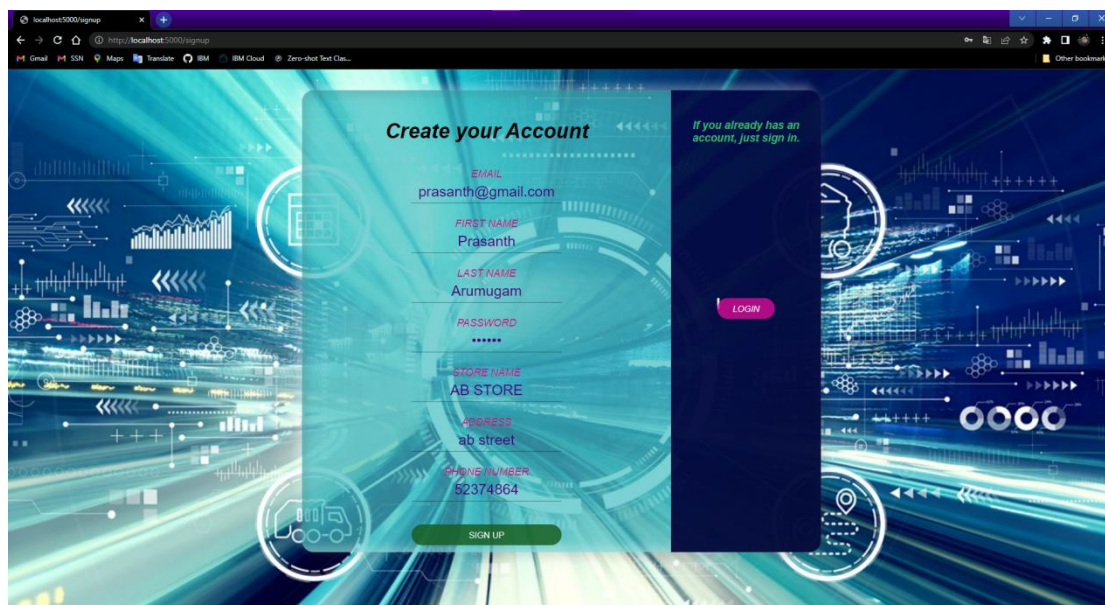
C:\Users\SSN\Desktop\SSN College\IBM\Project Development Phase\Sprint 3>docker run -p 5000:5000 sprint3
file @IBMConnection object at 0x7f8b23ce0790
Connecting Successful.....
* Serving Flask app "app"
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 127-736-908

```

Cover Page:



Signup Page:



Storing the Data into Database:

IBM Db2 on Cloud

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

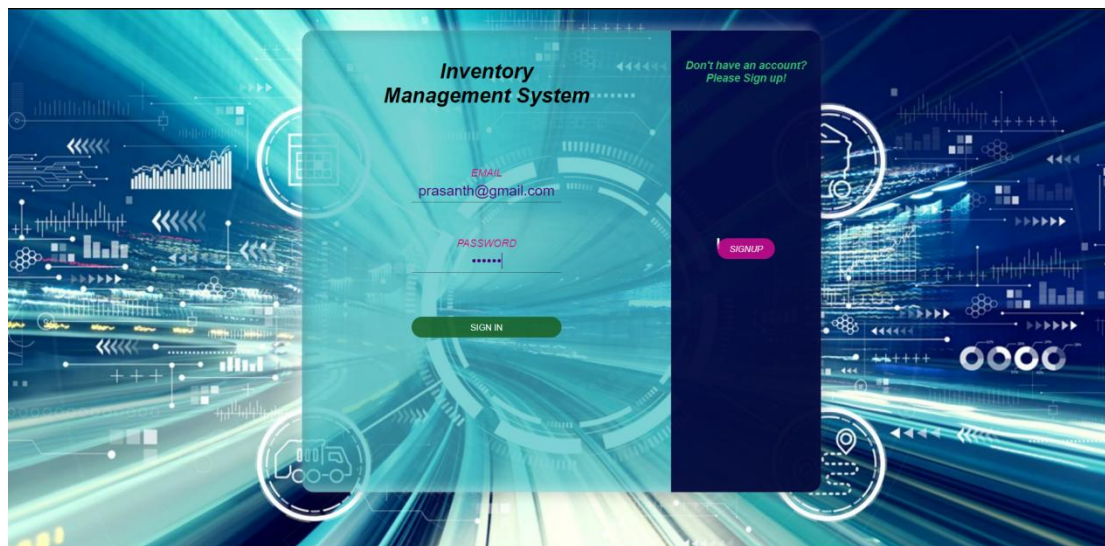
MMV18931.RETAILERS

Back

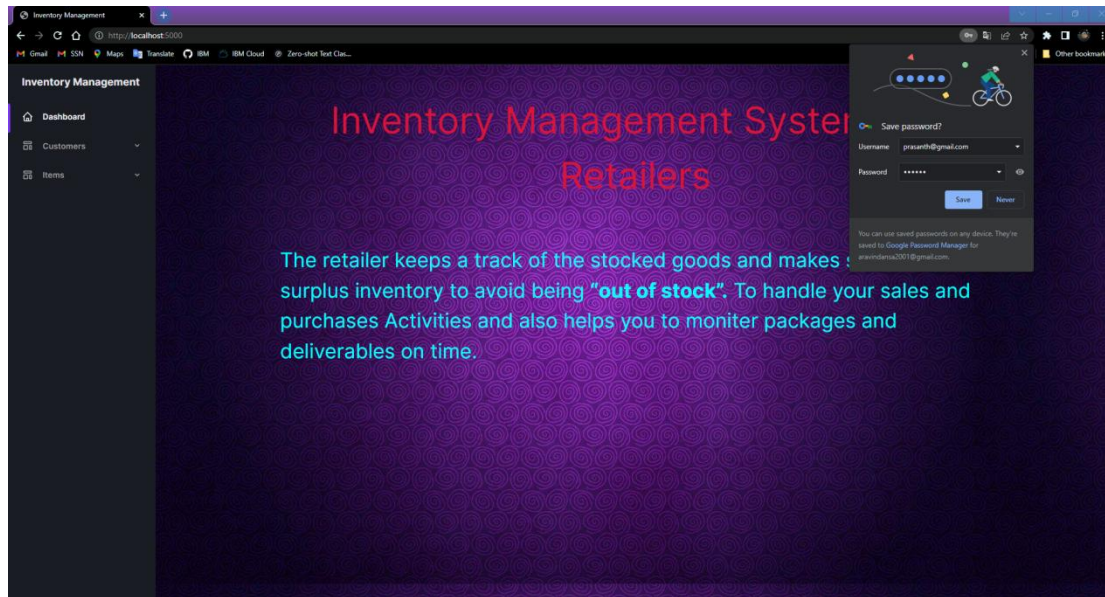
Export to CSV

RETAILER_ID	EMAIL	PASSWORD	FIRST_NAME	LAST_NAME	STORE_NAME	ADDRESS	PHONE_NUMBER
1	aravindan@gmail.com	@1234	Aravindan	S	ABC	chennai	6379528754
2	prasanth@gmail.com	123456	Prasanth	Arumugam	AB STORE	ab street	52374864

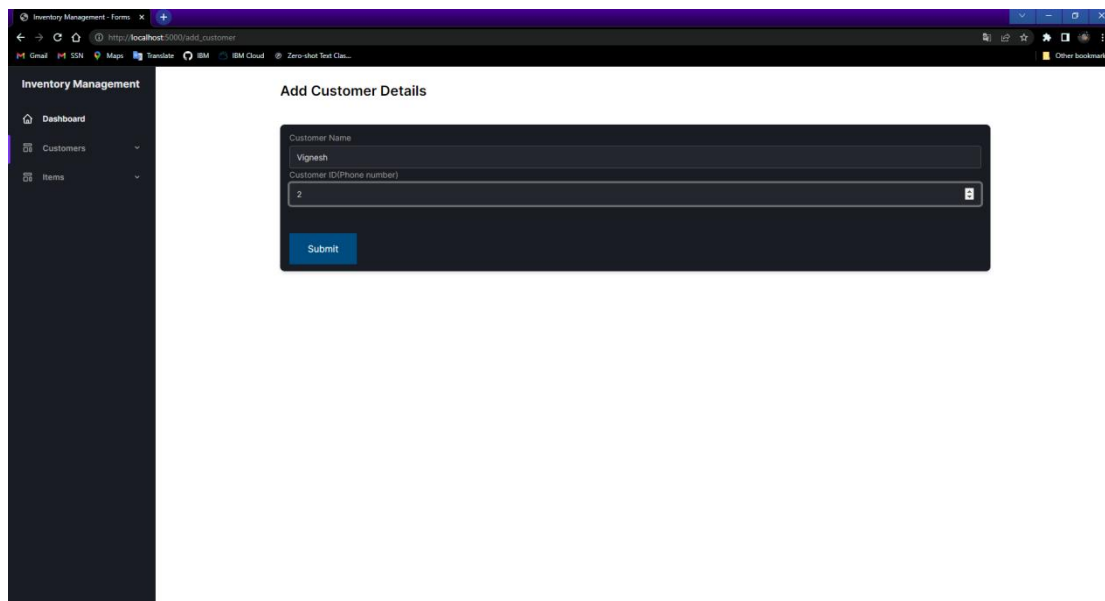
Login Page:



Home:



Add Customer:



Storing In Database:

IBM Db2 on Cloud

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

MMV18931.CUSTOMER Back

Export to CSV

CUSTOMER_ID	RETAILER_ID	CUSTOMER_NAME
1	1	prasanth
2	2	Vignesh

View Customers:

Inventory Management - Tables

Dashboard Customers Items

View Customers

Customers

CUSTOMER NAME	CUSTOMER ID(PHONE)
prasanth	1
Vignesh	2

Add Items:

Inventory Management - Forms

http://localhost:5000/add_item

Inventory Management

- Dashboard
- Customers
- Items

Add Item Details

Item Name
Biscuit

Price
10

Submit

Store In Database:

IBM Db2 on Cloud

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

MMV18931.ITEMS

Back

Export to CSV

ITEM_ID	RETAILER_ID	ITEM_NAME	PRICE	LEFT_OUT
2	1	chocolate	10	5
3	1	chocolate	10	0
4	1	apple	25	5
5	2	Biscuit	10	0

View Items:

The screenshot shows a web browser window with the URL `http://localhost:5000/view_item`. The application has a dark-themed sidebar on the left with the title 'Inventory Management'. The sidebar contains a 'Dashboard' link and two expandable menus: 'Customers' and 'Items'. The 'Items' menu is expanded, showing 'Inventory' as the selected option. Below the sidebar, the main content area is titled 'View Items' and displays a table with the following data:

ITEM ID	ITEM NAME	PRICE	LEFT OUT
2	chocolate	10	5
3	chocolate	10	0
4	apple	25	5
5	Biscuit	10	0

Add Inventory:

The screenshot shows the same web browser window, but the URL is `http://localhost:5000/add_inventory`. The sidebar is identical, but the 'Inventory' option in the 'Items' menu is highlighted. The main content area is titled 'Add Inventory Details' and contains a form with the following fields:

- Item Name:** A dropdown menu with 'Biscuit' selected.
- Quantity:** A text input field containing the value '10'.
- Stock Date:** A date picker field showing '27-11-2022'.
- Submit:** A blue button at the bottom of the form.

Store Into Database:

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The main content area displays the table 'MMV18931.INVENTORY'. A table structure is shown with columns: ITEM_ID, QUANTITY, and STOCK_DATE. The table contains three rows of data. A 'Back' button is in the top right, and an 'Export to CSV' button is in the bottom right.

ITEM_ID	QUANTITY	STOCK_DATE
2	5	2022-11-16
4	10	2022-11-17
5	10	2022-11-27

View Inventory:

The screenshot shows a web application titled 'View Inventory'. On the left is a sidebar menu for 'Inventory Management' with options: Dashboard, Customers, Items, Inventory (selected), and Sale. The main content area has a form with the following fields:

- Item Name: A dropdown menu with 'chocolate' selected.
- Start Date: A date input field with the placeholder 'dd-mm-yyyy'.
- End Date: A date input field with the placeholder 'dd-mm-yyyy'.

Below the form is an 'Inventory' table with the following data:

ITEM NAME	QUANTITY	DATE
Biscuit	10	2022-11-27

Add Sale:

The screenshot shows a web browser window with the URL `http://localhost:5000/add_sale`. The page title is "Add Sale". On the left, there is a sidebar menu for "Inventory Management" with options: Dashboard, Customers, Items, Inventory, and Sale (selected). Below the menu are buttons for "Add Sale" and "View Sale". The main form contains three input fields: "Customer" (with a dropdown menu showing "Vignesh"), "Item Name" (with a dropdown menu), and "Quantity". Below these fields is a blue button labeled "Add More". Underneath the form is a table with the following data:

ITEM NAME	PRICE	QUANTITY	TOTAL
Biscuit	10	4	40

Below the table, it says "Amount to be PAID :" followed by a yellow box containing the number "40". At the bottom of the form is a large blue button labeled "Submit".

Store into Database:

The screenshot shows the IBM Db2 console interface. At the top, there is a message: "Error: Please check network connectivity then try again." Below this, the table name "MMV18931.SALE" is displayed. To the right of the table name are buttons for "Back" and "Export to CSV". The table has three columns: "SALE_ID", "SALE_DATE", and "CUSTOMER_ID". The data rows are as follows:

SALE_ID	SALE_DATE	CUSTOMER_ID
1	2022-11-17	1
2	2022-11-17	1
3	2022-11-17	2

The screenshot shows the IBM Db2 console interface. At the top, there is a message: "Error: Please check network connectivity then try again." Below this, the table name "MMV18931.SALE_ITEMS" is displayed. To the right of the table name are buttons for "Back" and "Export to CSV". The table has three columns: "SALE_ID", "QUANTITY", and "ITEM_ID". The data rows are as follows:

SALE_ID	QUANTITY	ITEM_ID
1	2	4
1	3	4
3	4	5

View sale:

Inventory Management - Forms

http://localhost:5000/view_sale

GmailSSNMapsTranslateIBMIBM CloudZero-shot Test Clac...

Other bookmarks

Inventory Management

Dashboard

Customers

Items

Inventory

Sale

View Sale

Fill the form

Start Date

14-11-2022

End Date

20-11-2022

Filter

Items

ITEM ID	ITEM NAME	SOLD QUANTITY	AMOUNT
---------	-----------	---------------	--------

Inventory Management - Forms

http://localhost:5000/view_sale

GmailSSNMapsTranslateIBMIBM CloudZero-shot Test Clac...

Other bookmarks

Inventory Management

Dashboard

Customers

Items

Inventory

Sale

View Sale

Fill the form

Start Date

dd-mm-yyyy

End Date

dd-mm-yyyy

Items

ITEM ID	ITEM NAME	SOLD QUANTITY	AMOUNT
4	apple	5	125
5	Biscuit	4	40