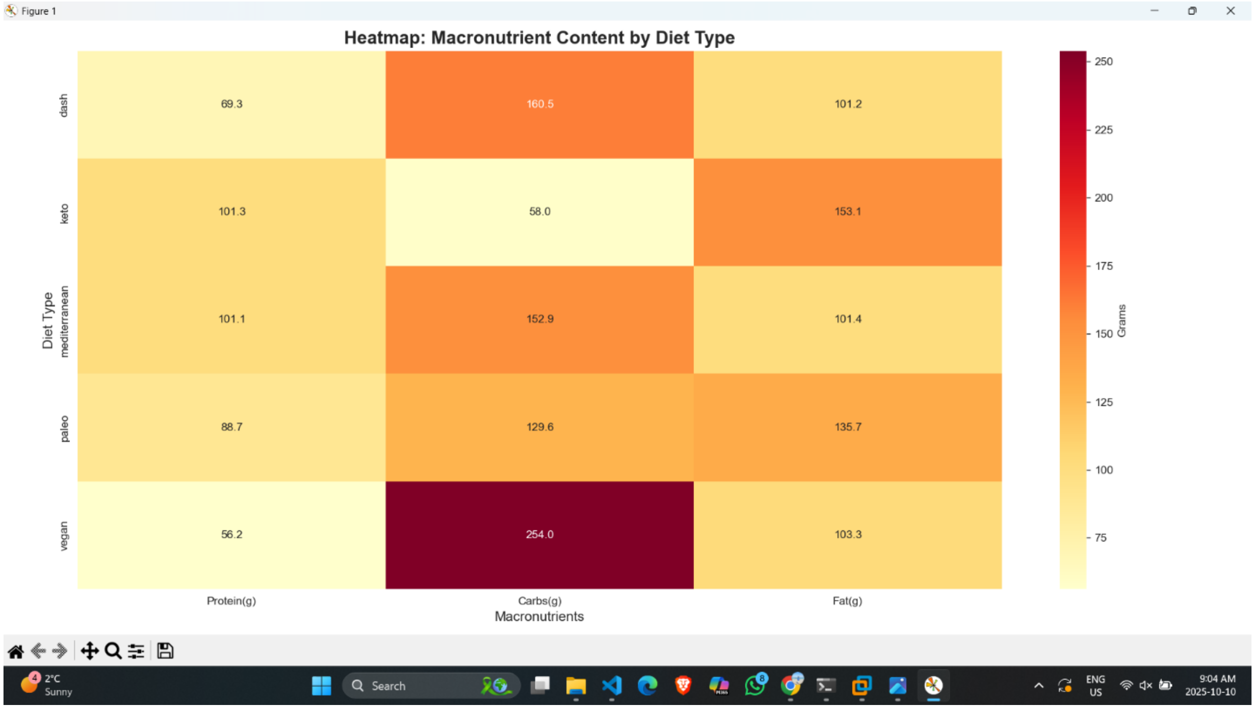
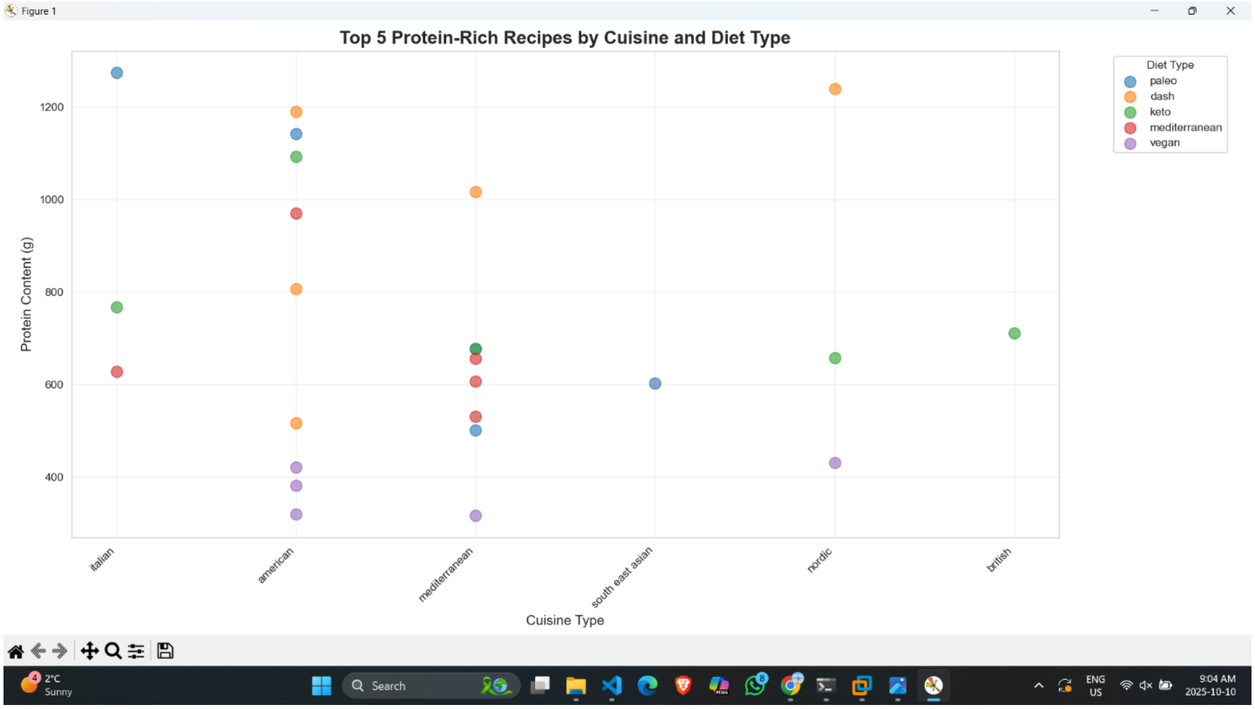
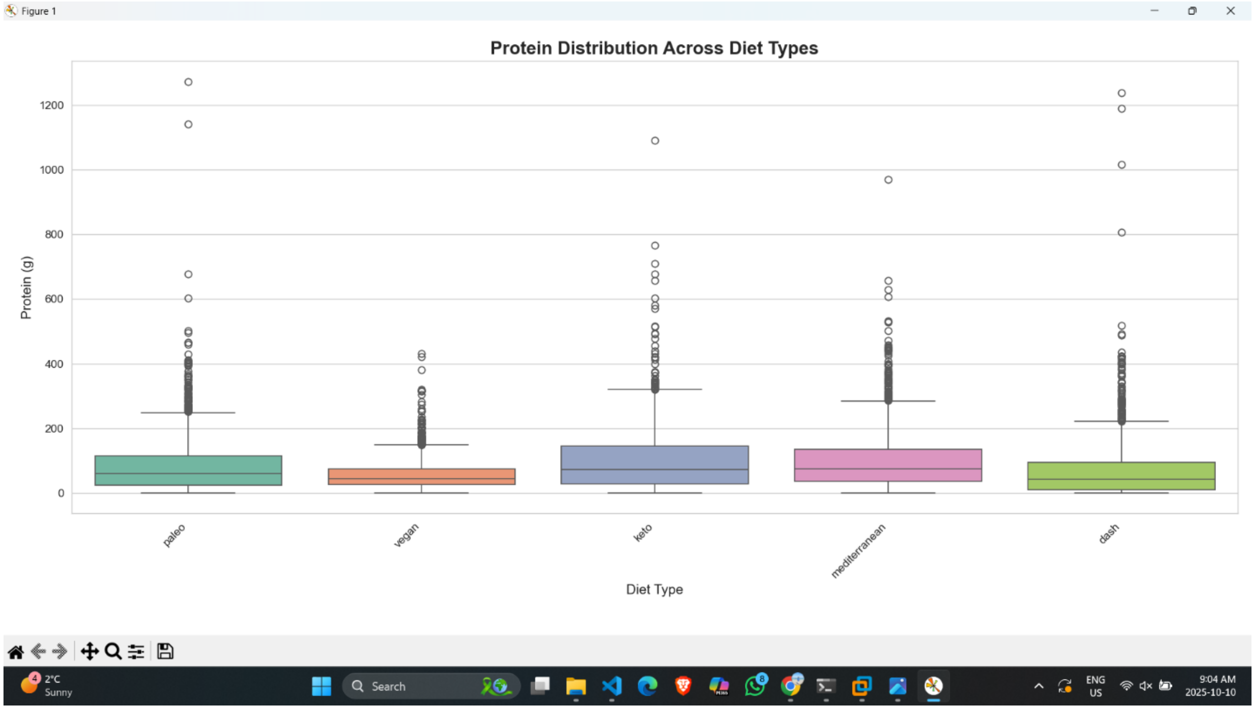
All files such as Docker Files, Py files and the GitHub actions will be in a ZIP with each in a folder named after the task its for

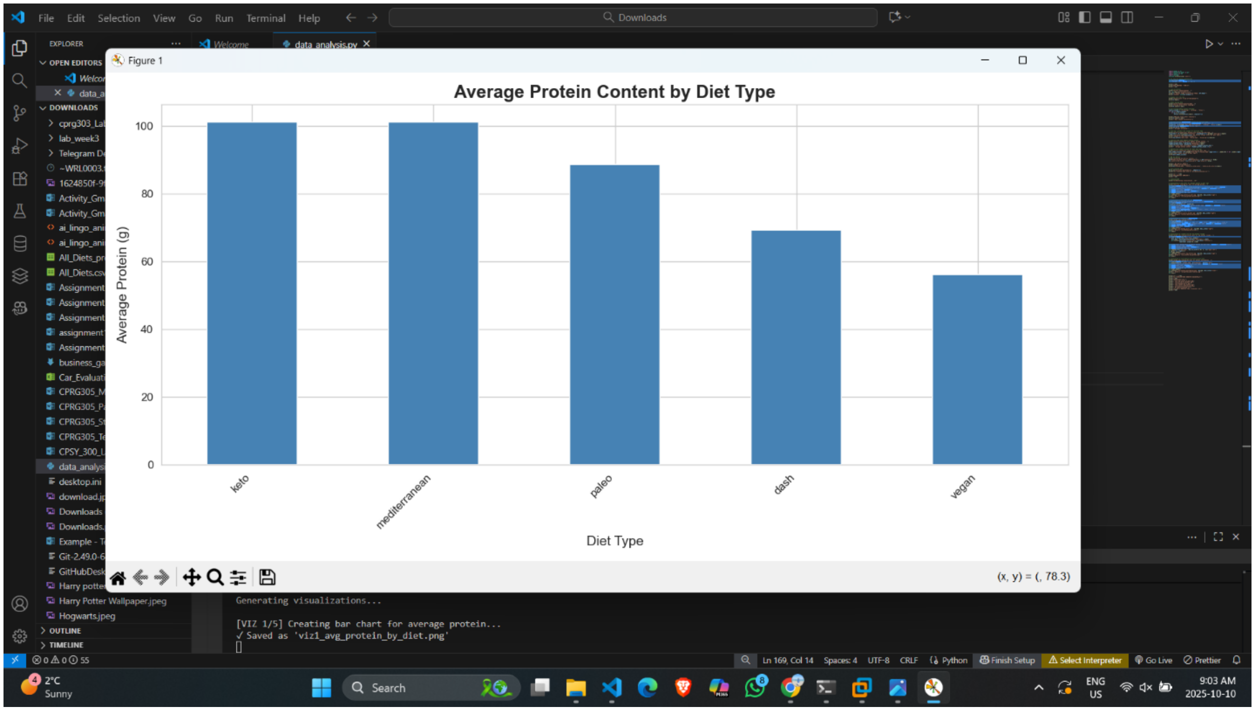
Task 1

Visualizations:

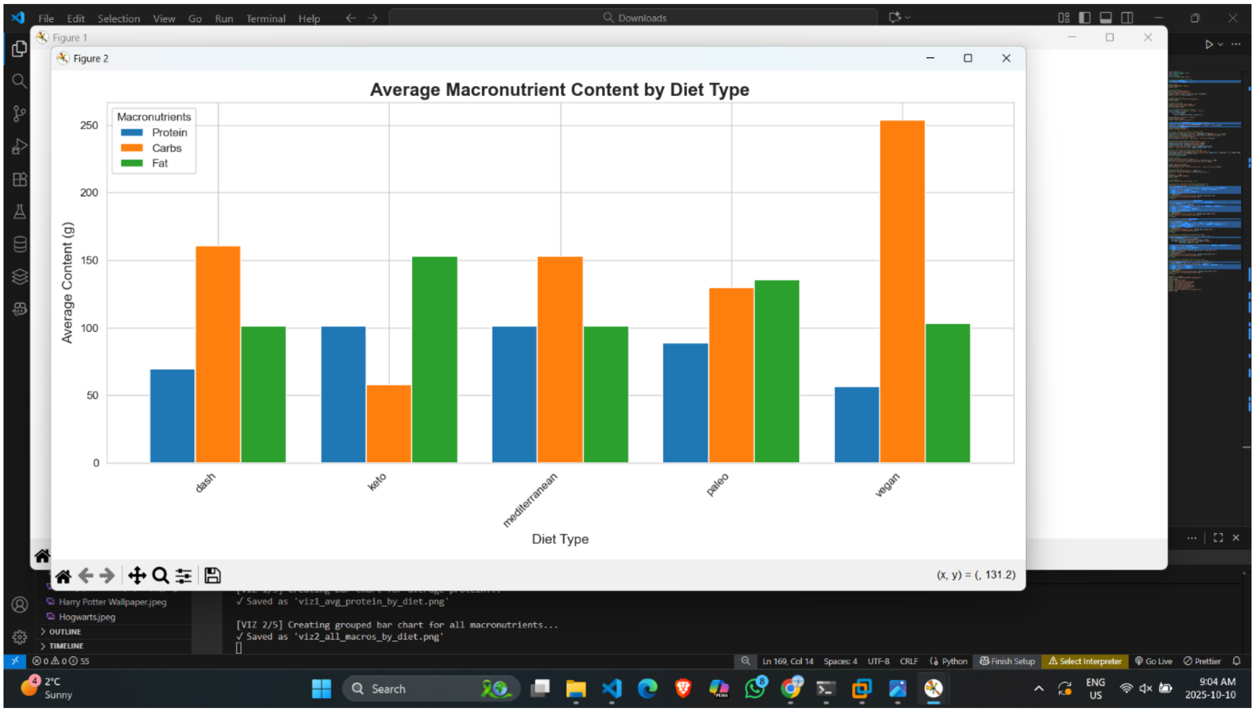






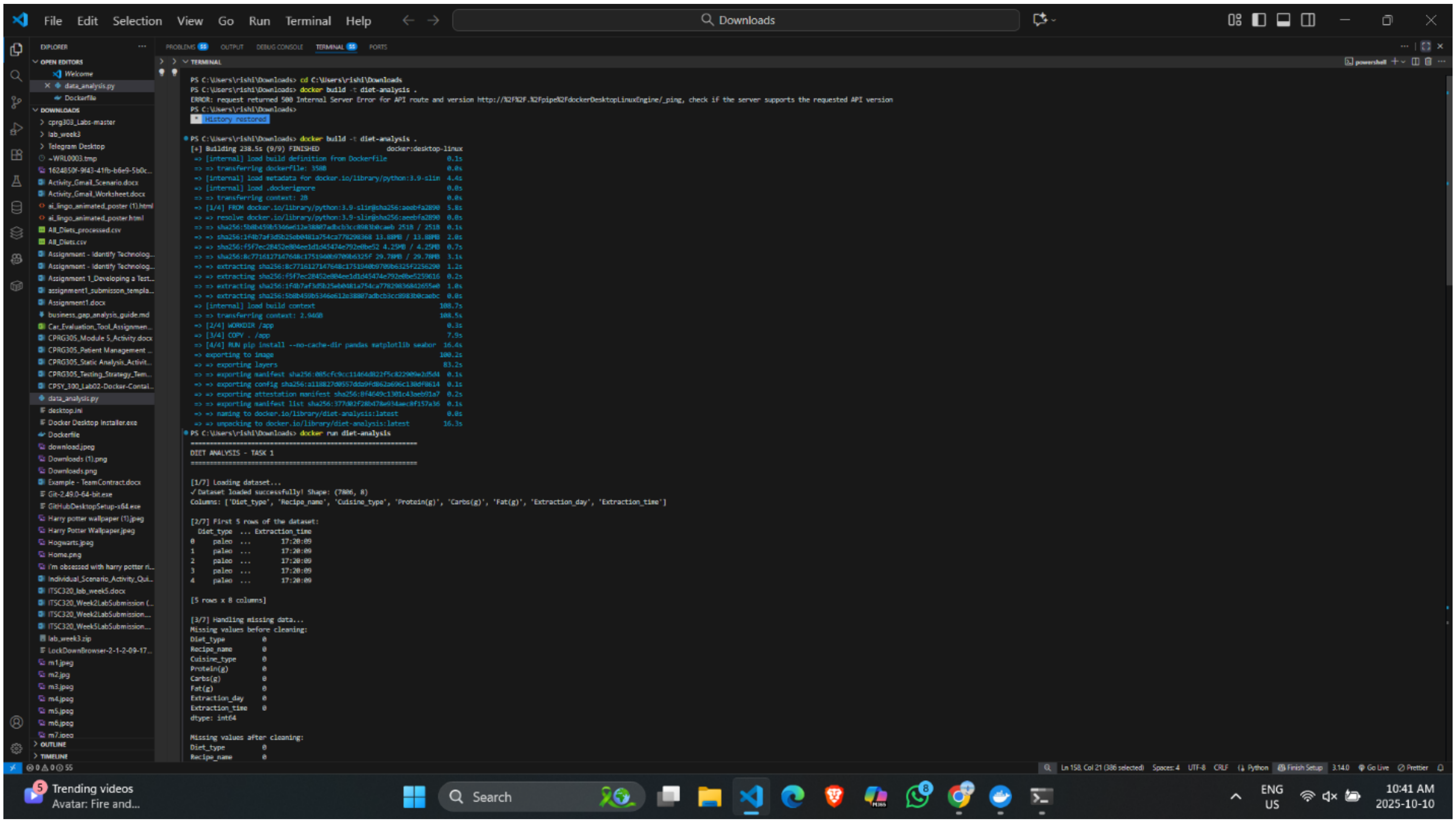


```
Generating visualizations...

[VIZ 1/5] Creating bar chart for average protein...
✓ Saved as 'viz1_avg_protein_by_diet.png'
```

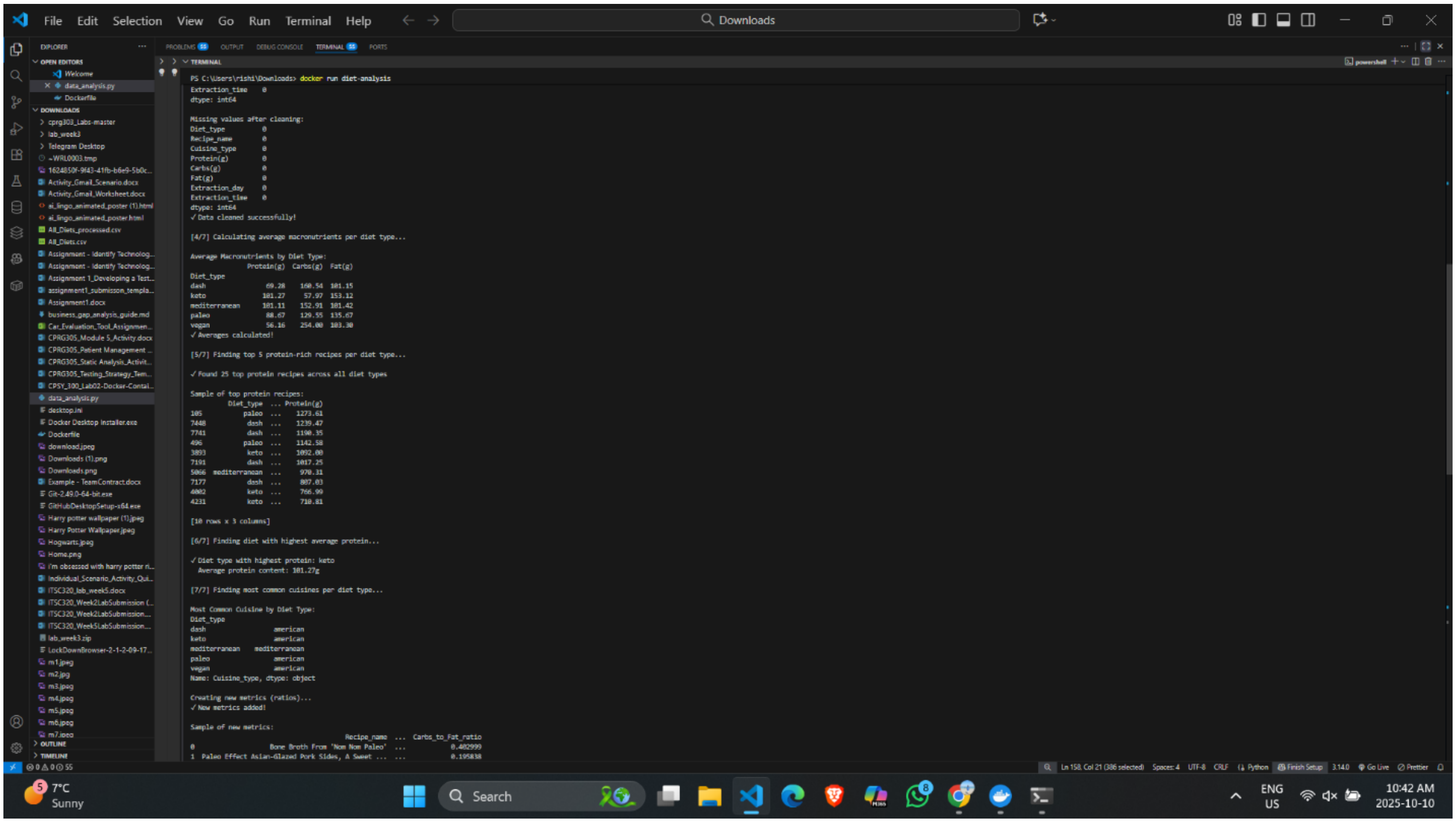


```
[VIZ 1/5] Creating bar chart for average protein...
✓ Saved as 'viz1_avg_protein_by_diet.png'

[VIZ 2/5] Creating grouped bar chart for all macronutrients...
✓ Saved as 'viz2_all_macros_by_diet.png'
```

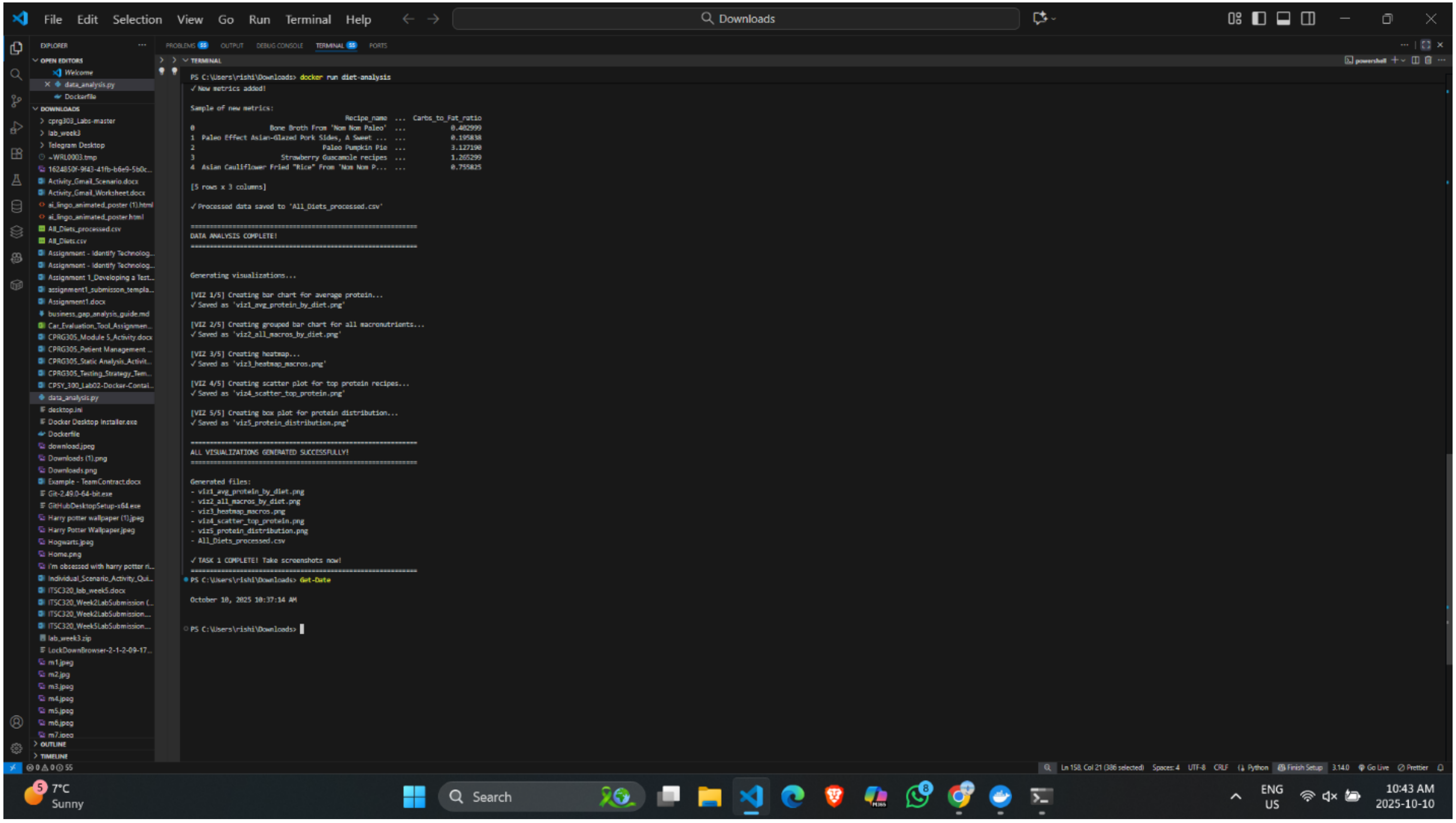
Task 2:



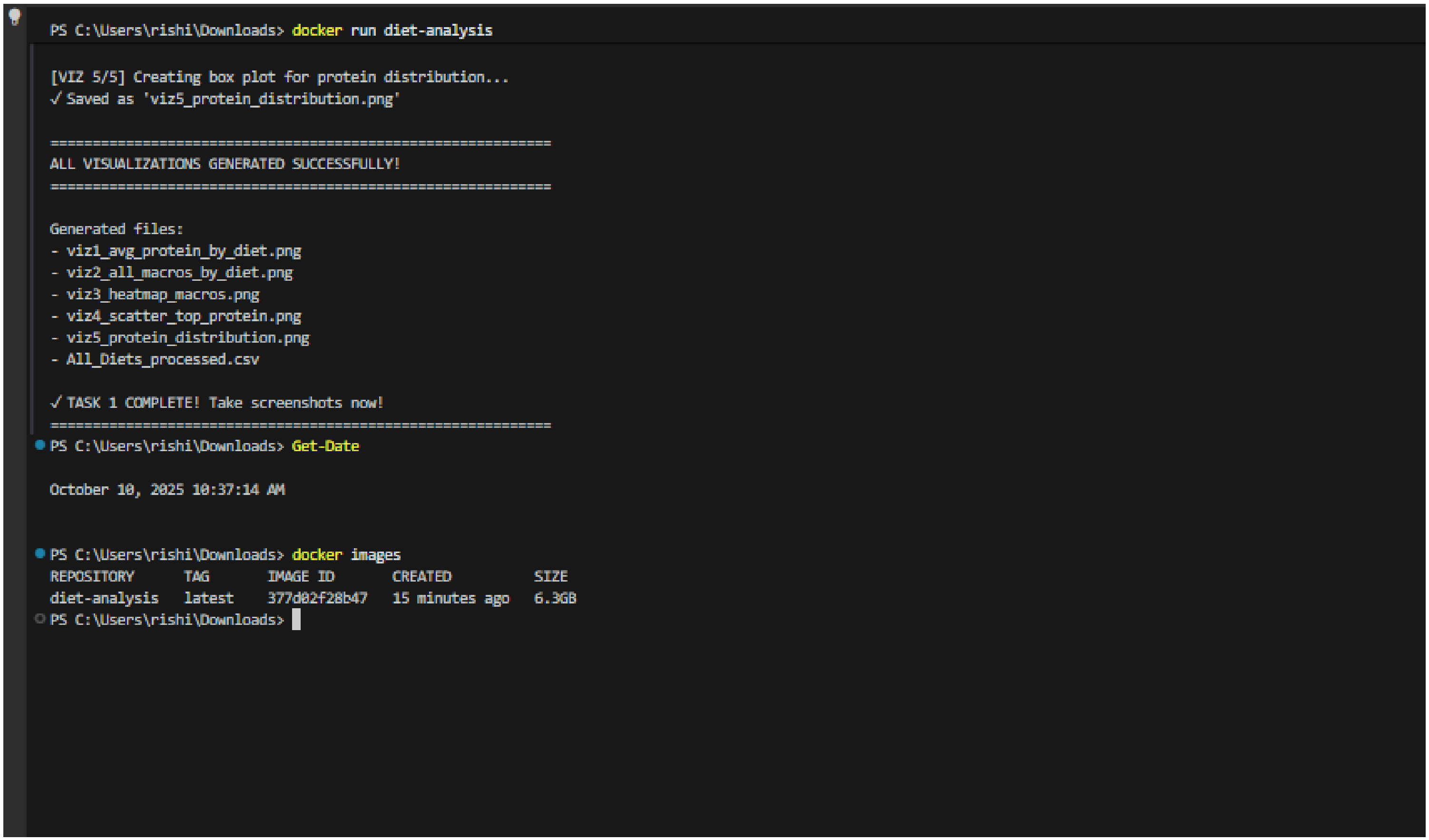
```
PS C:\Users\rishi\Downloads> cd C:\Users\rishi\Downloads
PS C:\Users\rishi\Downloads> docker build -t diet-analysis .
ERROR: request returned 500 Internal Server Error for API route and version http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/_ping, check if the server supports the requested API version
PS C:\Users\rishi\Downloads>
 * History restored

PS C:\Users\rishi\Downloads> docker build -t diet-analysis .
[+] Building 238.5s (9/9) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                        0.1s
 => => transferring dockerfile: 358B                                        0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim          4.4s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                             0.0s
 => [1/4] FROM docker.io/library/python:3.9-slim@sha256:aeebfa2890          5.8s
 => => resolve docker.io/library/python:3.9-slim@sha256:aeebfa2890          0.0s
 => => sha256:5b8b459b5346e612e38887adbcb3cc8983b0caeb 251B / 251B          0.1s
 => => sha256:1f4b7af3d5b25eb0481a754ca778298368 13.88MB / 13.88MB          2.0s
 => => sha256:f5f7ec28452e884ee1d1d45474e792e0be52 4.25MB / 4.25MB          0.7s
 => => sha256:8c7716127147648c1751940b9709b6325f 29.78MB / 29.78MB          3.1s
 => => extracting sha256:8c7716127147648c1751940b9709b6325f2256290          1.2s
 => => extracting sha256:f5f7ec28452e884ee1d1d45474e792e0be5259616          0.2s
 => => extracting sha256:1f4b7af3d5b25eb0481a754ca77829836842655e0          1.0s
 => => extracting sha256:5b8b459b5346e612e38887adbcb3cc8983b0caebc          0.0s
 => [internal] load build context                                         108.7s
 => => transferring context: 2.94GB                                       108.5s
 => [2/4] WORKDIR /app                                                      0.3s
 => [3/4] COPY . /app                                                       7.9s
 => [4/4] RUN pip install --no-cache-dir pandas matplotlib seabor          16.4s
 => exporting to image                                                    100.2s
 => => exporting layers                                                    83.2s
 => => exporting manifest sha256:085cfc9cc11464d822f5c822909e2d5d4          0.1s
 => => exporting config sha256:a118827d0557dda9fd862a696c138df8614          0.1s
 => => exporting attestation manifest sha256:8f4649c1301c43aeb01a7          0.2s
 => => exporting manifest list sha256:377d02f28bd78e934aec8f157a36          0.1s
 => => naming to docker.io/library/diet-analysis:latest                     0.0s
 => => unpacking to docker.io/library/diet-analysis:latest                 16.3s
PS C:\Users\rishi\Downloads> docker run diet-analysis
============================================================
DIET ANALYSIS - TASK 1
============================================================

[1/7] Loading dataset...
✓ Dataset loaded successfully! Shape: (7806, 8)
Columns: ['Diet_type', 'Recipe_name', 'Cuisine_type', 'Protein(g)', 'Carbs(g)', 'Fat(g)', 'Extraction_day', 'Extraction_time']

[2/7] First 5 rows of the dataset:
  Diet_type  ... Extraction_time
0     paleo  ...        17:20:09
1     paleo  ...        17:20:09
2     paleo  ...        17:20:09
3     paleo  ...        17:20:09
4     paleo  ...        17:20:09

[5 rows x 8 columns]

[3/7] Handling missing data...
Missing values before cleaning:
Diet_type          0
Recipe_name        0
Cuisine_type       0
Protein(g)         0
Carbs(g)           0
Fat(g)             0
Extraction_day     0
Extraction_time    0
dtype: int64

Missing values after cleaning:
Diet_type          0
Recipe_name        0
```



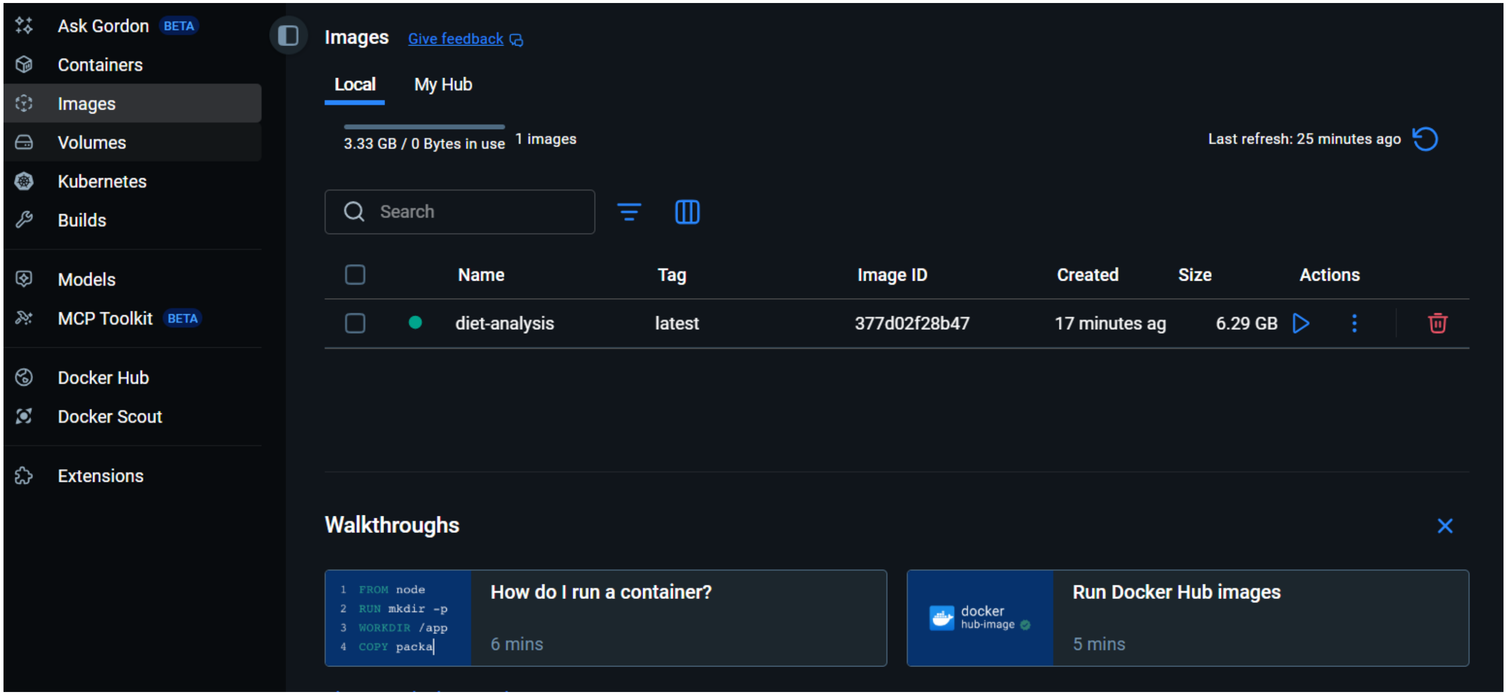
```
PS C:\Users\rishi\Downloads> docker run diet-analysis
Extraction_time    0
dtype: int64

Missing values after cleaning:
Diet_type          0
Recipe_name        0
Cuisine_type       0
Protein(g)         0
Carbs(g)           0
Fat(g)             0
Extraction_day     0
Extraction_time    0
dtype: int64
✓ Data cleaned successfully!

[4/7] Calculating average macronutrients per diet type...

Average Macronutrients by Diet Type:
               Protein(g)  Carbs(g)  Fat(g)
Diet_type
dash                69.28    160.54  101.15
keto               101.27     57.97  153.12
mediterranean      101.11    152.91  101.42
paleo               88.67    129.55  135.67
vegan               56.16    254.00  103.30
✓ Averages calculated!

[5/7] Finding top 5 protein-rich recipes per diet type...

✓ Found 25 top protein recipes across all diet types

Sample of top protein recipes:
          Diet_type  ... Protein(g)
105           paleo  ...    1273.61
7448           dash  ...    1239.47
7741           dash  ...    1190.35
496           paleo  ...    1142.58
3893           keto  ...    1092.00
7191           dash  ...    1017.25
5066  mediterranean  ...     970.31
7177           dash  ...     807.03
4002           keto  ...     766.99
4231           keto  ...     710.81

[10 rows x 3 columns]

[6/7] Finding diet with highest average protein...

✓ Diet type with highest protein: keto
  Average protein content: 101.27g

[7/7] Finding most common cuisines per diet type...

Most Common Cuisine by Diet Type:
Diet_type
dash                  american
keto                  american
mediterranean    mediterranean
paleo                 american
vegan                 american
Name: Cuisine_type, dtype: object

Creating new metrics (ratios)...
✓ New metrics added!

Sample of new metrics:
                                       Recipe_name  ...  Carbs_to_Fat_ratio
0                   Bone Broth From 'Nom Nom Paleo'  ...            0.402999
1  Paleo Effect Asian-Glazed Pork Sides, A Sweet ...  ...            0.195838
```



```
PS C:\Users\rishi\Downloads> docker run diet-analysis
✓ New metrics added!

Sample of new metrics:
                                       Recipe_name  ...  Carbs_to_Fat_ratio
0                   Bone Broth From 'Nom Nom Paleo'  ...            0.402999
1  Paleo Effect Asian-Glazed Pork Sides, A Sweet ...  ...            0.195838
2                                  Paleo Pumpkin Pie  ...            3.127190
3                       Strawberry Guacamole recipes  ...            1.265299
4  Asian Cauliflower Fried "Rice" From 'Nom Nom P...  ...            0.755825

[5 rows x 3 columns]

✓ Processed data saved to 'All_Diets_processed.csv'

============================================================
DATA ANALYSIS COMPLETE!
============================================================


Generating visualizations...

[VIZ 1/5] Creating bar chart for average protein...
✓ Saved as 'viz1_avg_protein_by_diet.png'

[VIZ 2/5] Creating grouped bar chart for all macronutrients...
✓ Saved as 'viz2_all_macros_by_diet.png'

[VIZ 3/5] Creating heatmap...
✓ Saved as 'viz3_heatmap_macros.png'

[VIZ 4/5] Creating scatter plot for top protein recipes...
✓ Saved as 'viz4_scatter_top_protein.png'

[VIZ 5/5] Creating box plot for protein distribution...
✓ Saved as 'viz5_protein_distribution.png'

============================================================
ALL VISUALIZATIONS GENERATED SUCCESSFULLY!
============================================================

Generated files:
- viz1_avg_protein_by_diet.png
- viz2_all_macros_by_diet.png
- viz3_heatmap_macros.png
- viz4_scatter_top_protein.png
- viz5_protein_distribution.png
- All_Diets_processed.csv

✓ TASK 1 COMPLETE! Take screenshots now!
============================================================
PS C:\Users\rishi\Downloads> Get-Date

October 10, 2025 10:37:14 AM


PS C:\Users\rishi\Downloads>
```



```
PS C:\Users\rishi\Downloads> docker run diet-analysis

[VIZ 5/5] Creating box plot for protein distribution...
✓ Saved as 'viz5_protein_distribution.png'

============================================================
ALL VISUALIZATIONS GENERATED SUCCESSFULLY!
============================================================

Generated files:
- viz1_avg_protein_by_diet.png
- viz2_all_macros_by_diet.png
- viz3_heatmap_macros.png
- viz4_scatter_top_protein.png
- viz5_protein_distribution.png
- All_Diets_processed.csv

✓ TASK 1 COMPLETE! Take screenshots now!
============================================================
PS C:\Users\rishi\Downloads> Get-Date

October 10, 2025 10:37:14 AM


PS C:\Users\rishi\Downloads> docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
diet-analysis   latest    377d02f28b47   15 minutes ago   6.3GB
PS C:\Users\rishi\Downloads>
```



Images

3.33 GB / 0 Bytes in use 1 images

Last refresh: 25 minutes ago

		Name	Tag	Image ID	Created	Size	Actions
□	●	diet-analysis	latest	377d02f28b47	17 minutes ag	6.29 GB	

Walkthroughs

```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

How do I run a container?

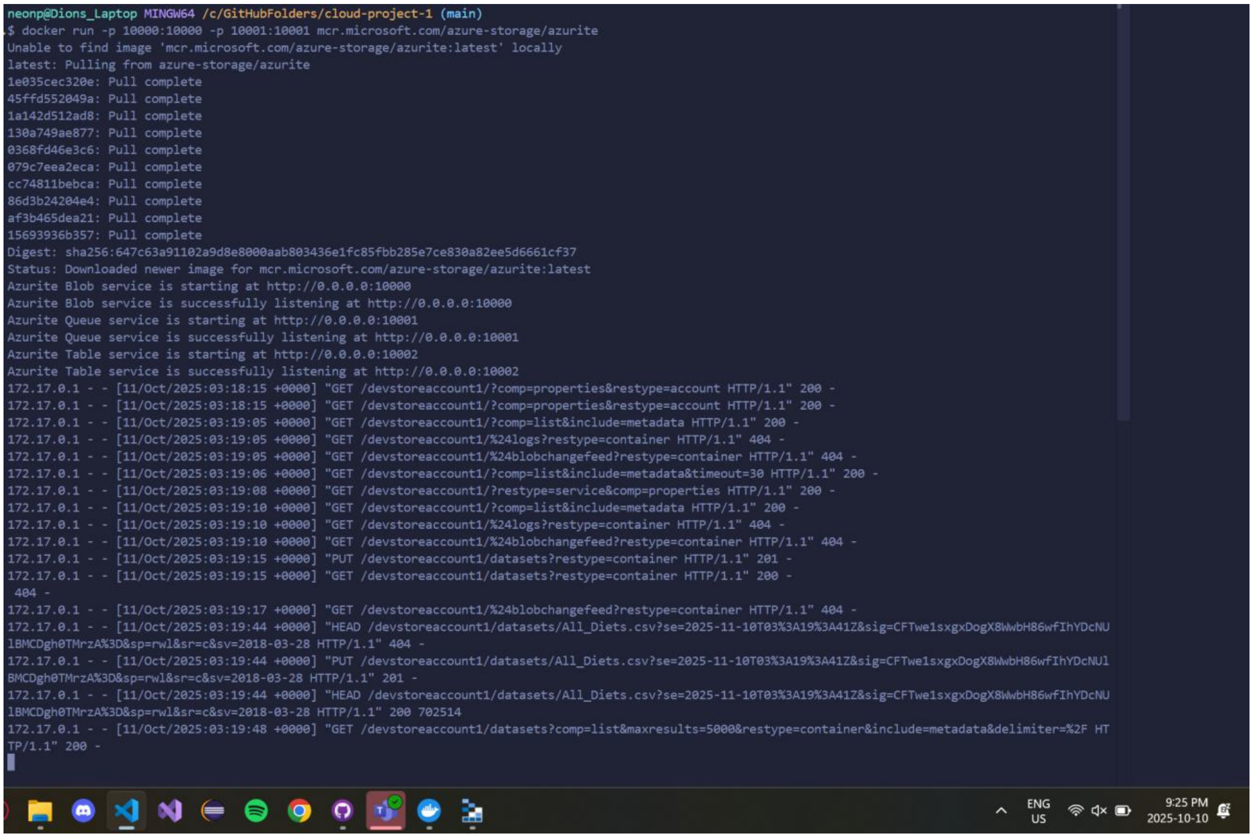
6 mins

Run Docker Hub images

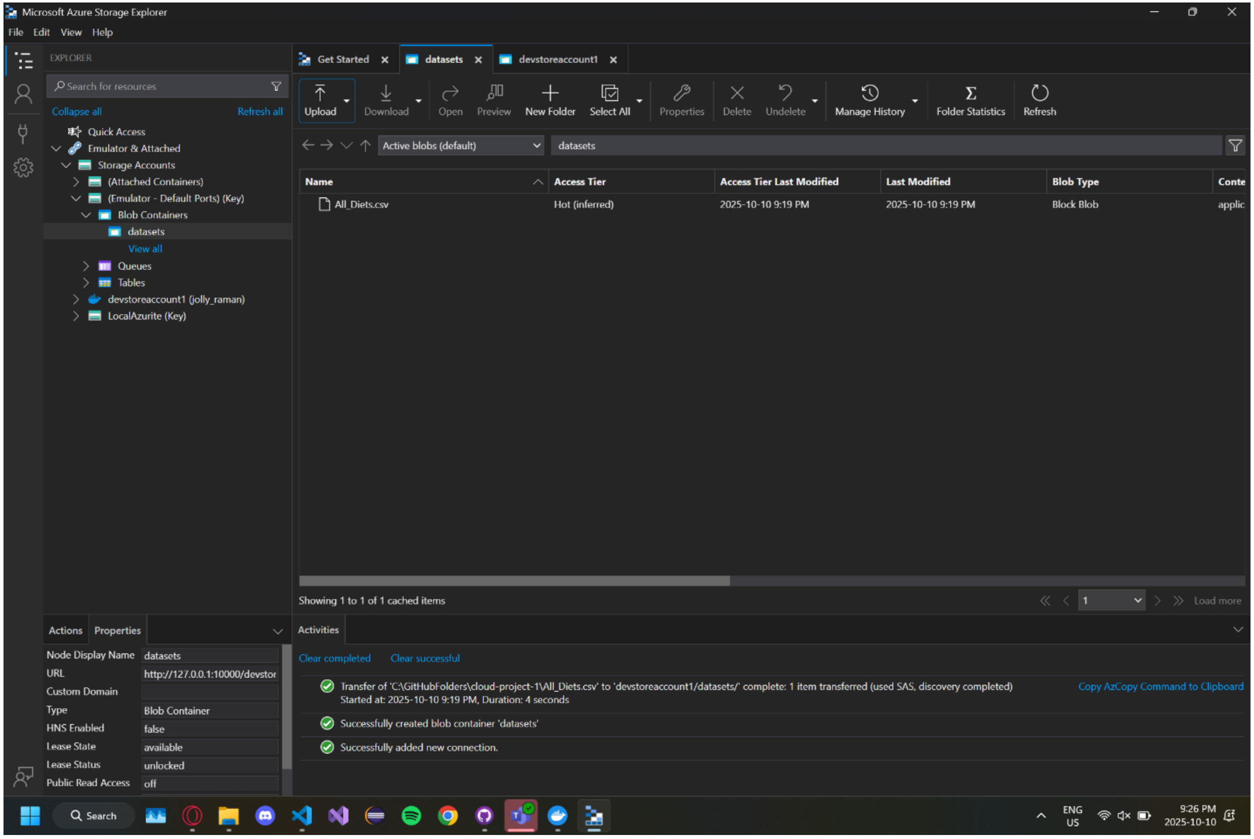
5 mins

Task 3:

The first screenshot shows it running but also me connection with Azure Storage Explorer



```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker run -p 10000:10000 -p 10001:10001 mcr.microsoft.com/azure-storage/azurite
Unable to find image 'mcr.microsoft.com/azure-storage/azurite:latest' locally
latest: Pulling from azure-storage/azurite
1e035cec320e: Pull complete
45ffd552049a: Pull complete
1a142d512ad8: Pull complete
130a749ae877: Pull complete
0368fd46e3c6: Pull complete
079c7eea2eca: Pull complete
cc74811bebca: Pull complete
86d3b24204e4: Pull complete
af3b465dea21: Pull complete
15693936b357: Pull complete
Digest: sha256:647c63a91102a9d8e8000aab803436e1fc85fbb285e7ce830a82ee5d6661cf37
Status: Downloaded newer image for mcr.microsoft.com/azure-storage/azurite:latest
Azurite Blob service is starting at http://0.0.0.0:10000
Azurite Blob service is successfully listening at http://0.0.0.0:10000
Azurite Queue service is starting at http://0.0.0.0:10001
Azurite Queue service is successfully listening at http://0.0.0.0:10001
Azurite Table service is starting at http://0.0.0.0:10002
Azurite Table service is successfully listening at http://0.0.0.0:10002
172.17.0.1 - - [11/Oct/2025:03:18:15 +0000] "GET /devstoreaccount1/?comp=properties&restype=account HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:18:15 +0000] "GET /devstoreaccount1/?comp=properties&restype=account HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:19:05 +0000] "GET /devstoreaccount1/?comp=list&include=metadata HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:19:05 +0000] "GET /devstoreaccount1/%24logs?restype=container HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:05 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:06 +0000] "GET /devstoreaccount1/?comp=list&include=metadata&timeout=30 HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:19:08 +0000] "GET /devstoreaccount1/?restype=service&comp=properties HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:19:10 +0000] "GET /devstoreaccount1/?comp=list&include=metadata HTTP/1.1" 200 -
172.17.0.1 - - [11/Oct/2025:03:19:10 +0000] "GET /devstoreaccount1/%24logs?restype=container HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:10 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:15 +0000] "PUT /devstoreaccount1/datasets?restype=container HTTP/1.1" 201 -
172.17.0.1 - - [11/Oct/2025:03:19:15 +0000] "GET /devstoreaccount1/datasets?restype=container HTTP/1.1" 200 -
 404 -
172.17.0.1 - - [11/Oct/2025:03:19:17 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:44 +0000] "HEAD /devstoreaccount1/datasets/All_Diets.csv?se=2025-11-10T03%3A19%3A41Z&sig=CFTwe1sxgxDogX8WwbH86wfIhYDcNU
lBMCDgh0TMrzA%3D&sp=rwl&sr=c&sv=2018-03-28 HTTP/1.1" 404 -
172.17.0.1 - - [11/Oct/2025:03:19:44 +0000] "PUT /devstoreaccount1/datasets/All_Diets.csv?se=2025-11-10T03%3A19%3A41Z&sig=CFTwe1sxgxDogX8WwbH86wfIhYDcNUl
BMCDgh0TMrzA%3D&sp=rwl&sr=c&sv=2018-03-28 HTTP/1.1" 201 -
172.17.0.1 - - [11/Oct/2025:03:19:44 +0000] "HEAD /devstoreaccount1/datasets/All_Diets.csv?se=2025-11-10T03%3A19%3A41Z&sig=CFTwe1sxgxDogX8WwbH86wfIhYDcNU
lBMCDgh0TMrzA%3D&sp=rwl&sr=c&sv=2018-03-28 HTTP/1.1" 200 702514
172.17.0.1 - - [11/Oct/2025:03:19:48 +0000] "GET /devstoreaccount1/datasets?comp=list&maxresults=5000&restype=container&include=metadata&delimiter=%2F HT
TP/1.1" 200 -
```



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Conte
All_Diets.csv	Hot (inferred)	2025-10-10 9:19 PM	2025-10-10 9:19 PM	Block Blob	applic

Showing 1 to 1 of 1 cached items

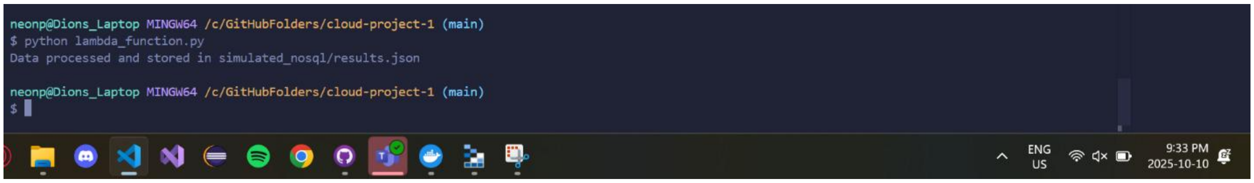
Node Display Name	datasets
URL	http://127.0.0.1:10000/devstor
Custom Domain	
Type	Blob Container
HNS Enabled	false
Lease State	available
Lease Status	unlocked
Public Read Access	off

✓ Transfer of 'C:\GitHubFolders\cloud-project-1\All_Diets.csv' to 'devstoreaccount1/datasets/' complete: 1 item transferred (used SAS, discovery completed)
Started at: 2025-10-10 9:19 PM, Duration: 4 seconds

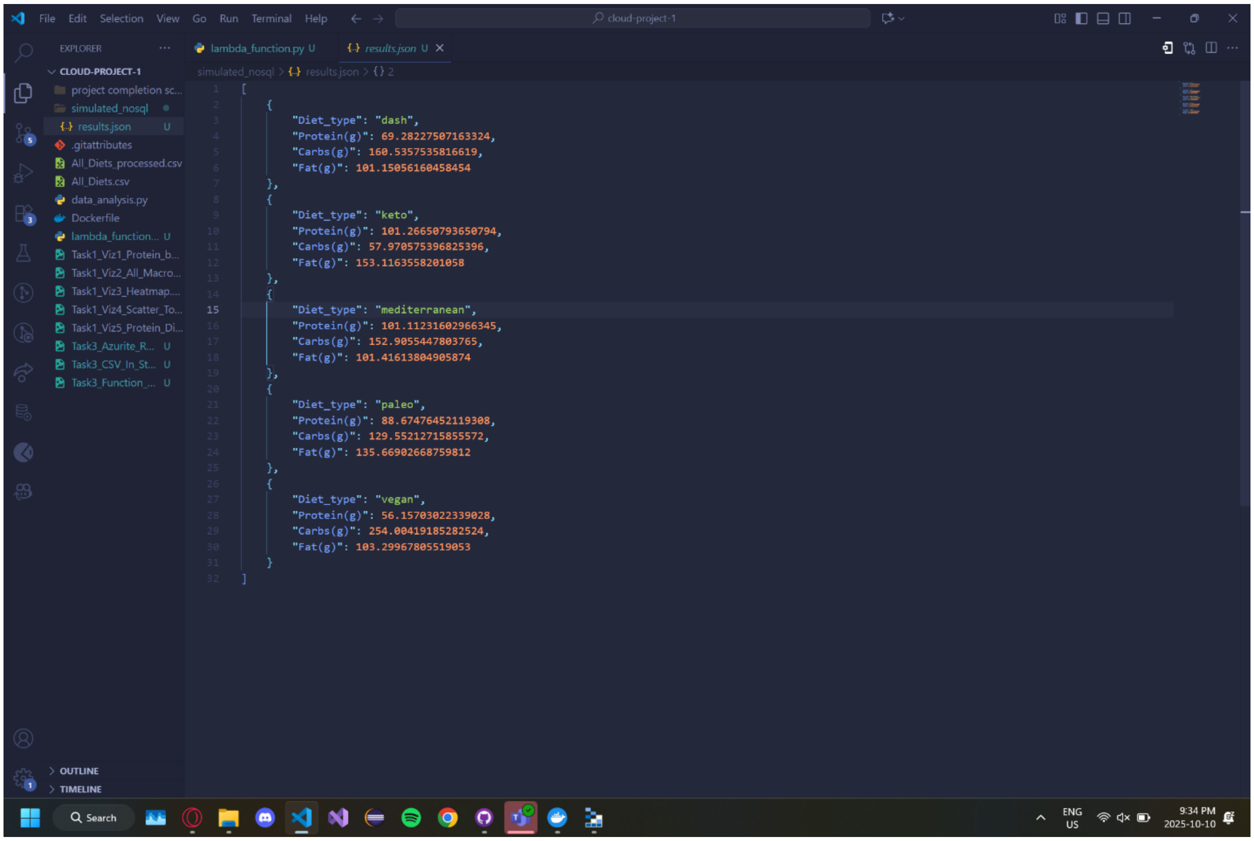
✓ Successfully created blob container 'datasets'

✓ Successfully added new connection.

The program just outputs this but creates the json file I sent below



```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ python lambda_function.py
Data processed and stored in simulated_nosql/results.json

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ 
```



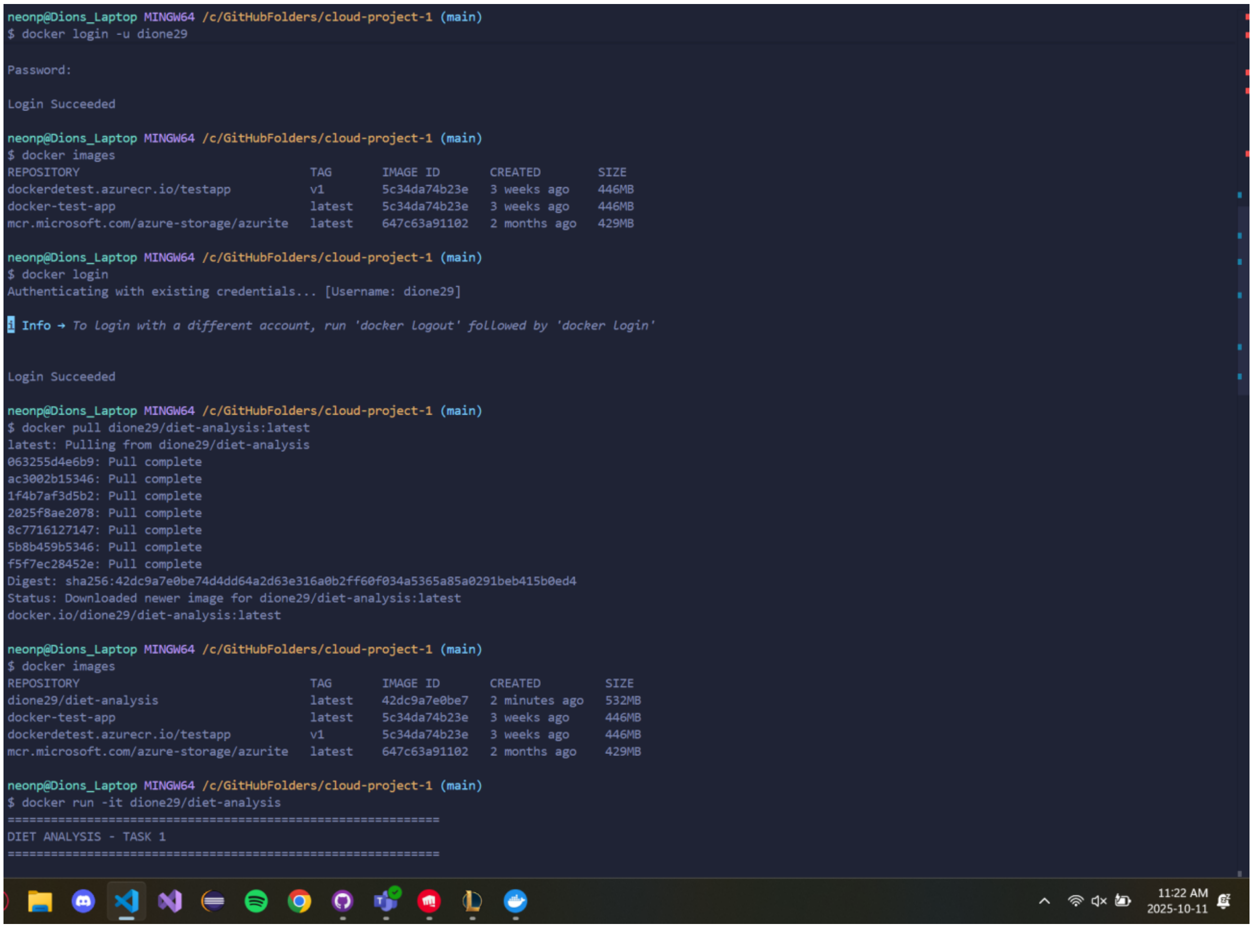
```json
[
    {
        "Diet_type": "dash",
        "Protein(g)": 69.28227507163324,
        "Carbs(g)": 160.5357535816619,
        "Fat(g)": 101.15056160458454
    },
    {
        "Diet_type": "keto",
        "Protein(g)": 101.26650793650794,
        "Carbs(g)": 57.970575396825396,
        "Fat(g)": 153.1163558201058
    },
    {
        "Diet_type": "mediterranean",
        "Protein(g)": 101.11231602966345,
        "Carbs(g)": 152.9055447803765,
        "Fat(g)": 101.41613804905874
    },
    {
        "Diet_type": "paleo",
        "Protein(g)": 88.67476452119308,
        "Carbs(g)": 129.55212715855572,
        "Fat(g)": 135.66902668759812
    },
    {
        "Diet_type": "vegan",
        "Protein(g)": 56.15703022339028,
        "Carbs(g)": 254.00419185282524,
        "Fat(g)": 103.29967805519053
    }
]
```

Task 3 Explanation:

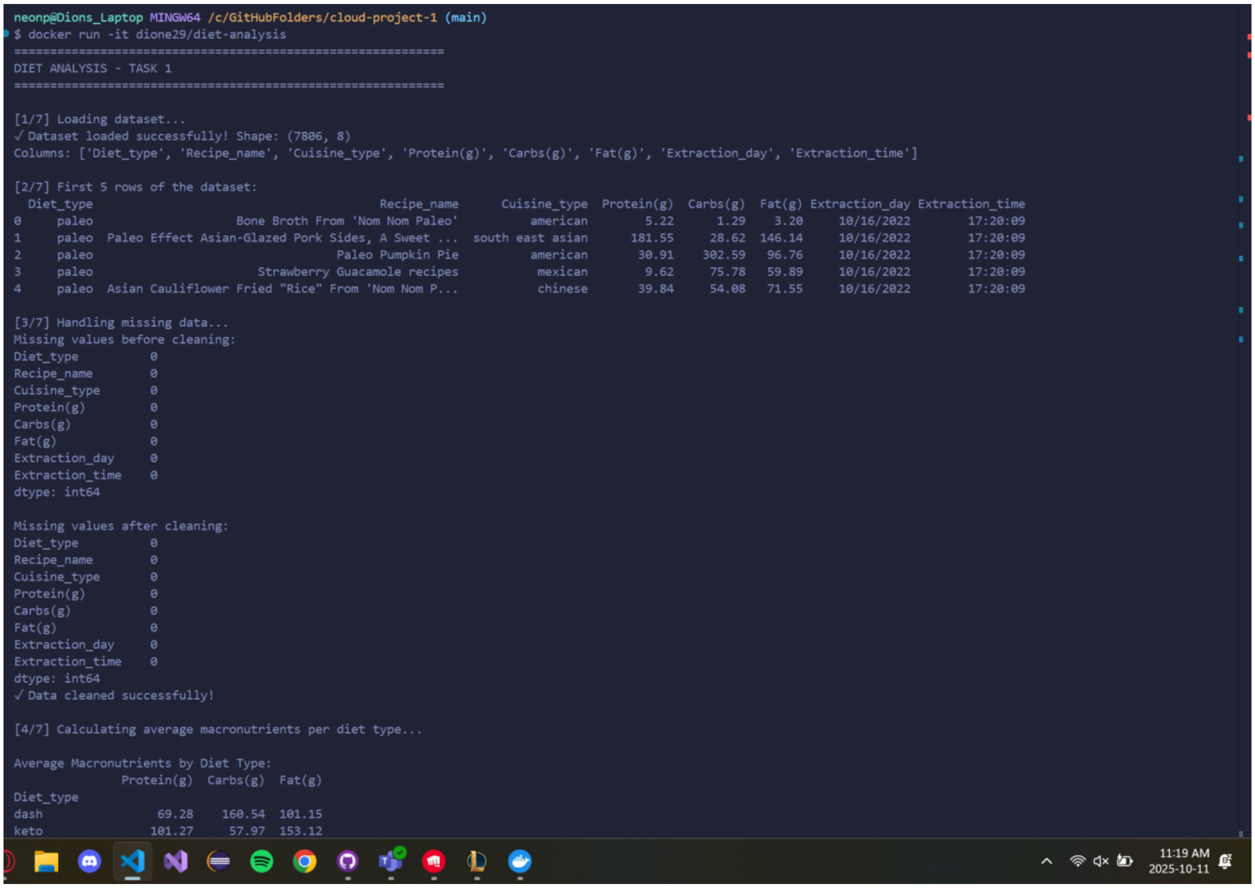
In this task, we used Azurite to create a local version of blob storage using both the pre-built docker image for azure-storage/azurite and Azure Storage Explorer.

We then uploaded the All_Diets.csv file to a blob container using Azure Storage Explorer and created a "serverless" function lambda_function.py to connect to the blob storage, read the dataset, process the data to get the results.json with each diet types average protein, carbs and fat. That represents our noSQL Database, this shows a local version of a serverless function that is stored in a simulated NoSQL Database

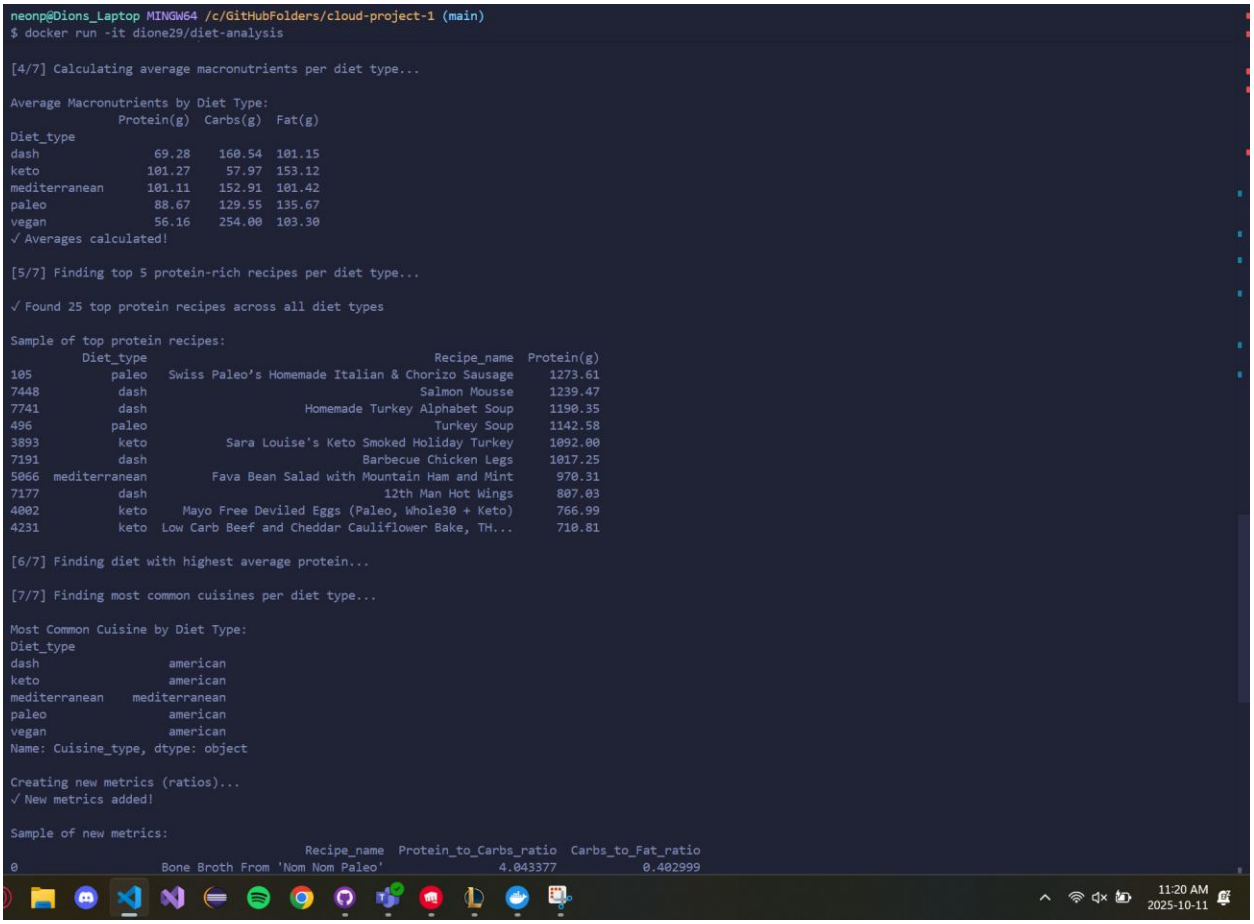
Task 4:



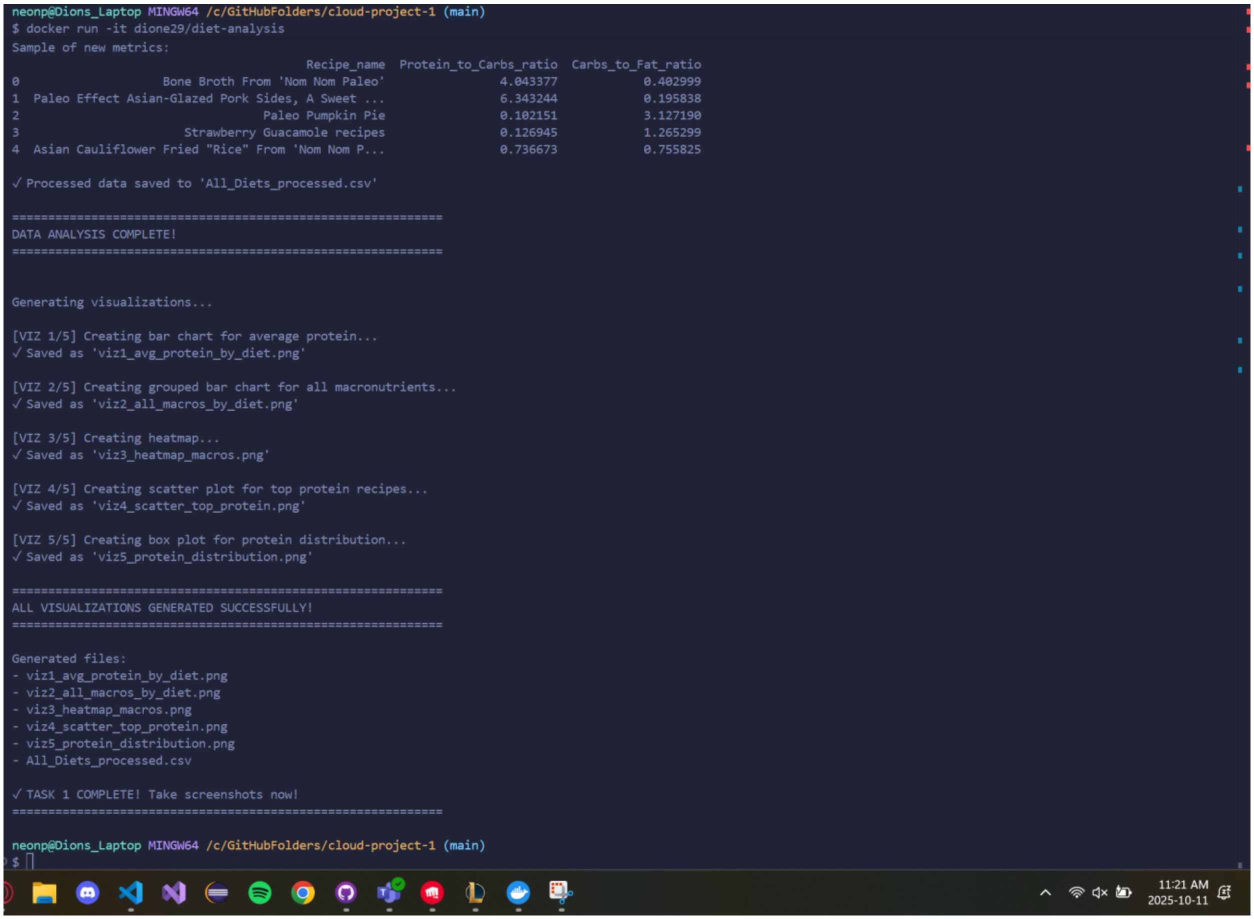
```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker login -u dione29

Password:

Login Succeeded

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker images
REPOSITORY                           TAG       IMAGE ID       CREATED        SIZE
dockerdetest.azurecr.io/testapp      v1        5c34da74b23e   3 weeks ago    446MB
docker-test-app                      latest    5c34da74b23e   3 weeks ago    446MB
mcr.microsoft.com/azure-storage/azurite   latest    647c63a91102   2 months ago   429MB

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker login
Authenticating with existing credentials... [Username: dione29]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'


Login Succeeded

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker pull dione29/diet-analysis:latest
latest: Pulling from dione29/diet-analysis
063255d4e6b9: Pull complete
ac3002b15346: Pull complete
1f4b7af3d5b2: Pull complete
2025f8ae2078: Pull complete
8c7716127147: Pull complete
5b8b459b5346: Pull complete
f5f7ec28452e: Pull complete
Digest: sha256:42dc9a7e0be74d4dd64a2d63e316a0b2ff60f034a5365a85a0291beb415b0ed4
Status: Downloaded newer image for dione29/diet-analysis:latest
docker.io/dione29/diet-analysis:latest

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker images
REPOSITORY                           TAG       IMAGE ID       CREATED          SIZE
dione29/diet-analysis                latest    42dc9a7e0be7   2 minutes ago    532MB
docker-test-app                      latest    5c34da74b23e   3 weeks ago      446MB
dockerdetest.azurecr.io/testapp      v1        5c34da74b23e   3 weeks ago      446MB
mcr.microsoft.com/azure-storage/azurite   latest    647c63a91102   2 months ago     429MB

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker run -it dione29/diet-analysis
============================================================
DIET ANALYSIS - TASK 1
============================================================
```



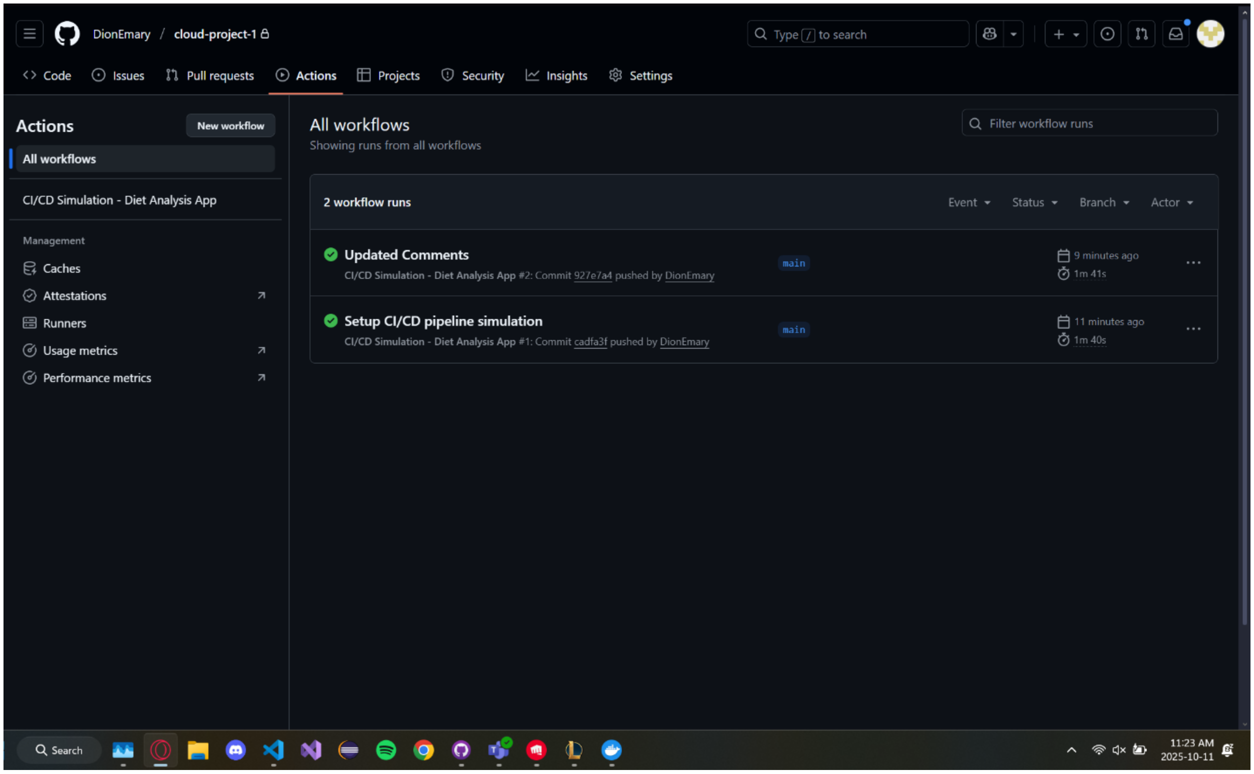
```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker run -it dione29/diet-analysis
============================================================
DIET ANALYSIS - TASK 1
============================================================

[1/7] Loading dataset...
✓ Dataset loaded successfully! Shape: (7806, 8)
Columns: ['Diet_type', 'Recipe_name', 'Cuisine_type', 'Protein(g)', 'Carbs(g)', 'Fat(g)', 'Extraction_day', 'Extraction_time']

[2/7] First 5 rows of the dataset:
  Diet_type                                         Recipe_name      Cuisine_type  Protein(g)  Carbs(g)  Fat(g) Extraction_day Extraction_time
0     paleo                    Bone Broth From 'Nom Nom Paleo'          american        5.22      1.29    3.20     10/16/2022        17:20:09
1     paleo  Paleo Effect Asian-Glazed Pork Sides, A Sweet ...  south east asian      181.55     28.62  146.14     10/16/2022        17:20:09
2     paleo                                  Paleo Pumpkin Pie          american       30.91    302.59   96.76     10/16/2022        17:20:09
3     paleo                       Strawberry Guacamole recipes           mexican        9.62     75.78   59.89     10/16/2022        17:20:09
4     paleo  Asian Cauliflower Fried "Rice" From 'Nom Nom P...           chinese       39.84     54.08   71.55     10/16/2022        17:20:09

[3/7] Handling missing data...
Missing values before cleaning:
Diet_type          0
Recipe_name        0
Cuisine_type       0
Protein(g)         0
Carbs(g)           0
Fat(g)             0
Extraction_day     0
Extraction_time    0
dtype: int64

Missing values after cleaning:
Diet_type          0
Recipe_name        0
Cuisine_type       0
Protein(g)         0
Carbs(g)           0
Fat(g)             0
Extraction_day     0
Extraction_time    0
dtype: int64
✓ Data cleaned successfully!

[4/7] Calculating average macronutrients per diet type...

Average Macronutrients by Diet Type:
               Protein(g)  Carbs(g)  Fat(g)
Diet_type
dash                69.28    160.54  101.15
keto               101.27     57.97  153.12
```



```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker run -it dione29/diet-analysis

[4/7] Calculating average macronutrients per diet type...

Average Macronutrients by Diet Type:
               Protein(g)  Carbs(g)  Fat(g)
Diet_type
dash                69.28    160.54  101.15
keto               101.27     57.97  153.12
mediterranean      101.11    152.91  101.42
paleo               88.67    129.55  135.67
vegan               56.16    254.00  103.30
✓ Averages calculated!

[5/7] Finding top 5 protein-rich recipes per diet type...

✓ Found 25 top protein recipes across all diet types

Sample of top protein recipes:
          Diet_type                                         Recipe_name  Protein(g)
105           paleo   Swiss Paleo’s Homemade Italian & Chorizo Sausage     1273.61
7448           dash                                      Salmon Mousse     1239.47
7741           dash                      Homemade Turkey Alphabet Soup     1190.35
496           paleo                                        Turkey Soup     1142.58
3893           keto           Sara Louise's Keto Smoked Holiday Turkey     1092.00
7191           dash                              Barbecue Chicken Legs     1017.25
5066  mediterranean         Fava Bean Salad with Mountain Ham and Mint      970.31
7177           dash                                12th Man Hot Wings      807.03
4002           keto     Mayo Free Deviled Eggs (Paleo, Whole30 + Keto)      766.99
4231           keto  Low Carb Beef and Cheddar Cauliflower Bake, TH...      710.81

[6/7] Finding diet with highest average protein...

[7/7] Finding most common cuisines per diet type...

Most Common Cuisine by Diet Type:
Diet_type
dash                  american
keto                  american
mediterranean    mediterranean
paleo                 american
vegan                 american
Name: Cuisine_type, dtype: object

Creating new metrics (ratios)...
✓ New metrics added!

Sample of new metrics:
                                         Recipe_name  Protein_to_Carbs_ratio  Carbs_to_Fat_ratio
0                    Bone Broth From 'Nom Nom Paleo'                4.043377            0.402999
```



```
neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ docker run -it dione29/diet-analysis
Sample of new metrics:
                                         Recipe_name  Protein_to_Carbs_ratio  Carbs_to_Fat_ratio
0                    Bone Broth From 'Nom Nom Paleo'                4.043377            0.402999
1  Paleo Effect Asian-Glazed Pork Sides, A Sweet ...                6.343244            0.195838
2                                  Paleo Pumpkin Pie                0.102151            3.127190
3                       Strawberry Guacamole recipes                0.126945            1.265299
4  Asian Cauliflower Fried "Rice" From 'Nom Nom P...                0.736673            0.755825

✓ Processed data saved to 'All_Diets_processed.csv'

============================================================
DATA ANALYSIS COMPLETE!
============================================================


Generating visualizations...

[VIZ 1/5] Creating bar chart for average protein...
✓ Saved as 'viz1_avg_protein_by_diet.png'

[VIZ 2/5] Creating grouped bar chart for all macronutrients...
✓ Saved as 'viz2_all_macros_by_diet.png'

[VIZ 3/5] Creating heatmap...
✓ Saved as 'viz3_heatmap_macros.png'

[VIZ 4/5] Creating scatter plot for top protein recipes...
✓ Saved as 'viz4_scatter_top_protein.png'

[VIZ 5/5] Creating box plot for protein distribution...
✓ Saved as 'viz5_protein_distribution.png'

============================================================
ALL VISUALIZATIONS GENERATED SUCCESSFULLY!
============================================================

Generated files:
- viz1_avg_protein_by_diet.png
- viz2_all_macros_by_diet.png
- viz3_heatmap_macros.png
- viz4_scatter_top_protein.png
- viz5_protein_distribution.png
- All_Diets_processed.csv

✓ TASK 1 COMPLETE! Take screenshots now!
============================================================

neonp@Dions_Laptop MINGW64 /c/GitHubFolders/cloud-project-1 (main)
$ 
```

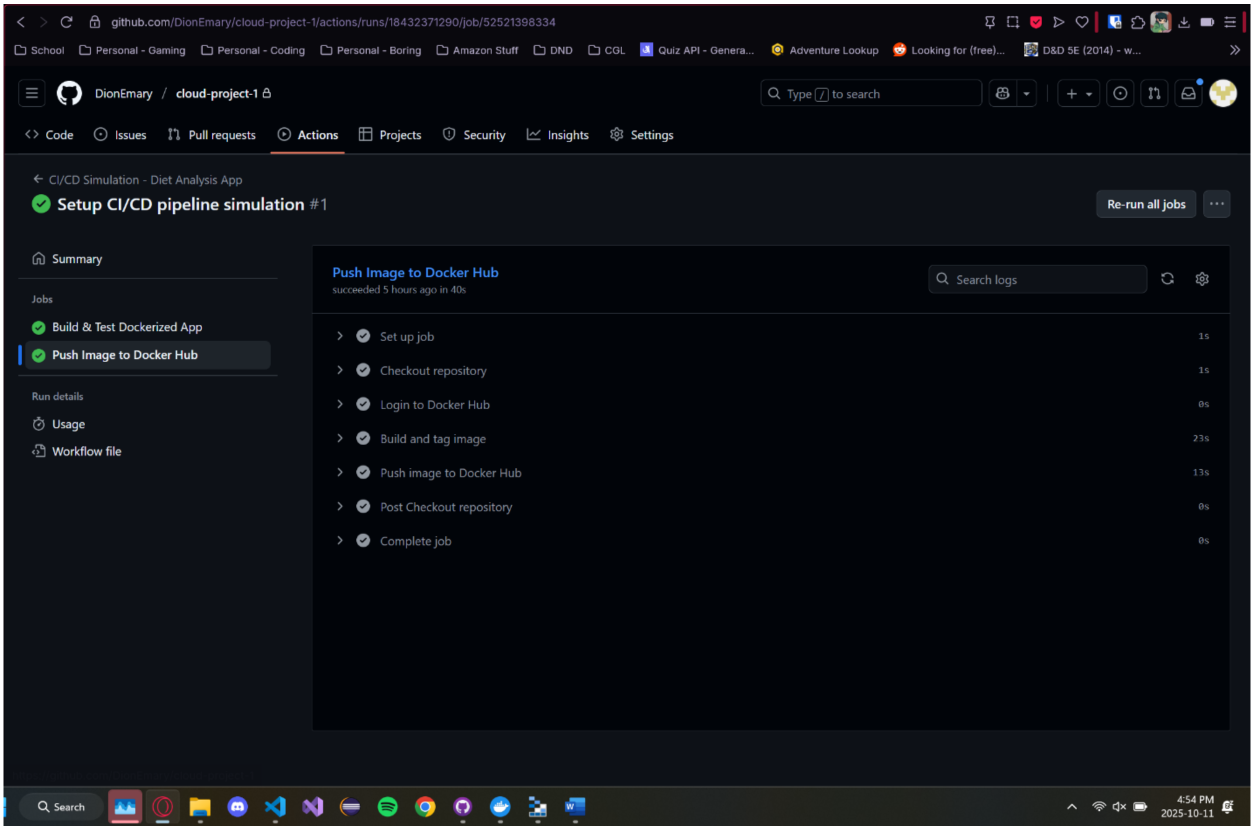


All workflows

Showing runs from all workflows

2 workflow runs

✓ **Updated Comments** CI/CD Simulation - Diet Analysis App #2: Commit 927e7a4 pushed by DionEmary	main	9 minutes ago 1m 41s
✓ **Setup CI/CD pipeline simulation** CI/CD Simulation - Diet Analysis App #1: Commit cadfa3f pushed by DionEmary	main	11 minutes ago 1m 40s

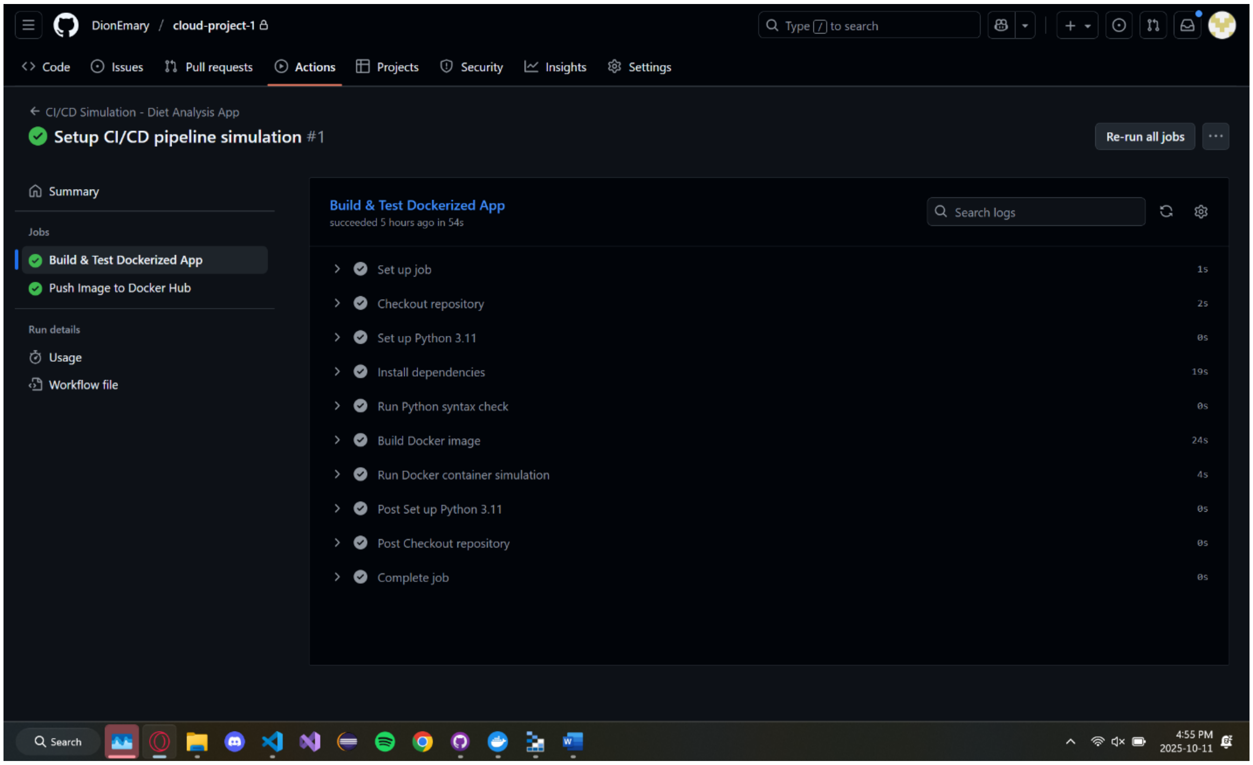


CI/CD Simulation - Diet Analysis App

Setup CI/CD pipeline simulation #1

Push Image to Docker Hub
succeeded 5 hours ago in 40s

```
Set up job                    1s
Checkout repository           1s
Login to Docker Hub           0s
Build and tag image          23s
Push image to Docker Hub     13s
Post Checkout repository      0s
Complete job                  0s
```



CI/CD Simulation - Diet Analysis App

Setup CI/CD pipeline simulation #1

Build & Test Dockerized App
succeeded 5 hours ago in 54s

```
Set up job                          1s
Checkout repository                 2s
Set up Python 3.11                  0s
Install dependencies               19s
Run Python syntax check             0s
Build Docker image                 24s
Run Docker container simulation     4s
Post Set up Python 3.11             0s
Post Checkout repository            0s
Complete job                        0s
```

The screenshots above show me pulling the Docker Image created BY the github actions and showing it running locally. But I also show the logs as well I can see.

Task 5:

For enhancing our project we went with two specific routes, one was to add multiple stages to our docker file, the other was improving the queries and searches made for Task 3 in the Lambda_function.py file. When it comes to the research done we found that for multi-stage docker builds, it is faster to split it into stages as in some cases we don't need to constantly rebuild the docker image every time we want to run it for example, in a CI/CD pipeline. Instead we can split it into multiple stages then have each stage after the other simply copy the others files that are already built rather than rerunning the whole build again. For multistage deployment we split it into two stages, one is the build phase that creates the directory with the dependencies, the other adds the project files and such to the actual docker image. This makes it so instead of having to rebuild the entire image every time, we can instead just build it once and then copy all the needed code after to runway faster. From testing it made our program execute way quicker in the GitHub actions, as it didn't need to build the entire image each push, instead it would copy the already build directory with the dependencies and then run the code inside of that copy.

As for our second improvement, we worked with the panda library to speed up how we process our data but also improved how we read the CSV file. From research we found two major issues and one smaller one that were hurting performance, the first one was that panda read all columns in the CSV file, rather than only the needed data. The second one was that when we processed the data, we were sorting them by their diet_type string, but we found you can use panda to convert it to a numeric value based on the string when reading. This helps improve the speed we can process data as it's easier for us to do all the data processing with numbers rather than strings. The final optimization is having panda pull the numbers at float32's rather than float64, as this halves the size of the data. Float64 can read long decimal places, but since we only go up to 2 decimal places, float32 can still read the data just fine while taking less storage. The improvements were done by adding two variables, usecols and dtypes, usecols contains the column names for each column we need to process data. This is used to tell panda it only needs those columns rather than all of them, shortening how long it takes to load the CSV file. The second one, dtypes is used to tell panda how to pull the data correctly, first it tells panda that it needs to create a numeric table for each value in diet type. What this does is that it assigns a number to each possible value so keto would be 0, vegan would be 1 and etc. This is then stored in a map so we can convert between number and string, this is used to convert it for data processing later on. This speeds up data processing as we can just use numeric values which are easier to sort by, then after merging the data, we can convert the diet type back to a string to get the same results. The second part of dtypes is that it tells panda it can pull the fat, protein and carb numbers as float32 rather than 64 which results in the same data, just less storage used thus speeding up how we pull the data and processing it too, even if small per number, it adds up quickly. We can expect these small changes to improve our execution speed by small amounts each time, but together it made a noticeable difference when testing it out, as it would execute noticeably faster (a few seconds) than the initial execution.