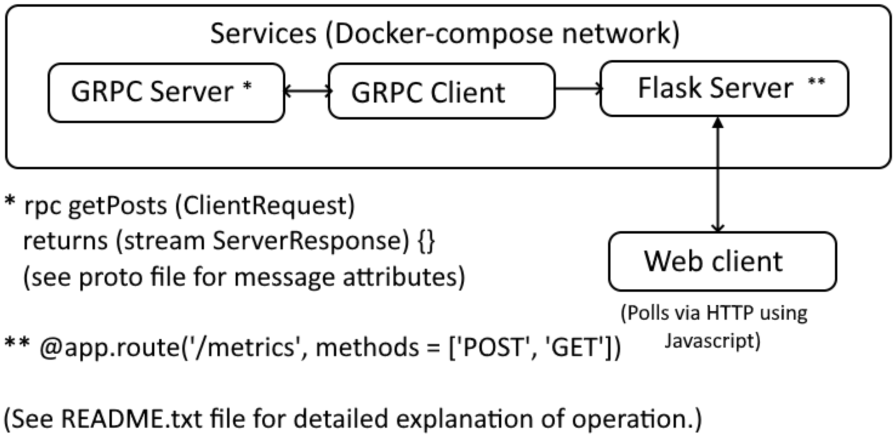
SOFT8026 - Data-driven Microservices
Assignment 1 Form

Instructions
Please complete the following form and include in the zip file you submit. Include screenshots / images in the appendices below the form.

Discussion of Architecture	
Put a diagram of your architecture opposite (e.g. paste an image); indicate the microservices, integration endpoints, messages being sent, etc.	 Services (Docker-compose network) GRPC Server * GRPC Client Flask Server ** Web client (Polls via HTTP using Javascript) * rpc getPosts (ClientRequest) returns (stream ServerResponse) {} (see proto file for message attributes) ** @app.route('/metrics', methods = ['POST', 'GET']) (See README.txt file for detailed explanation of operation.)
Why did you opt for the microservices you did? You could address topics from the lectures, e.g. bounded contexts, independent pipelines, etc.	The three services as shown above, each with their own container were sufficient to achieve the desired outcome. i.e. the simulation of a remote server streaming blog post data to a client. In a real-world scenario, it would seem appropriate to have a bounded context separating the GRPC server from the other two services as it would be streaming to the client service from somewhere remote. However, in this simulation, the whole of the docker-compose network can be seen as one bounded context in which the Flask server provides two endpoints for clients. One to render the webpage and another for the webpage to GET the latest metrics data via continuous polling. No data storage service / container was needed here as it was decided to have the GRPC client program store the streamed CSV data rows in memory as they arrive (using a dictionary). It then computes updated metrics and posts them via HTTP to the Flask service.
Checklist of requirements completed	**Delete as appropriate**
Do you have a Dockerfile per microservice that you have created?	YES
Do you have a gRPC proto file with a service, rpcs and messages for each microservice?	YES – see included proto file.

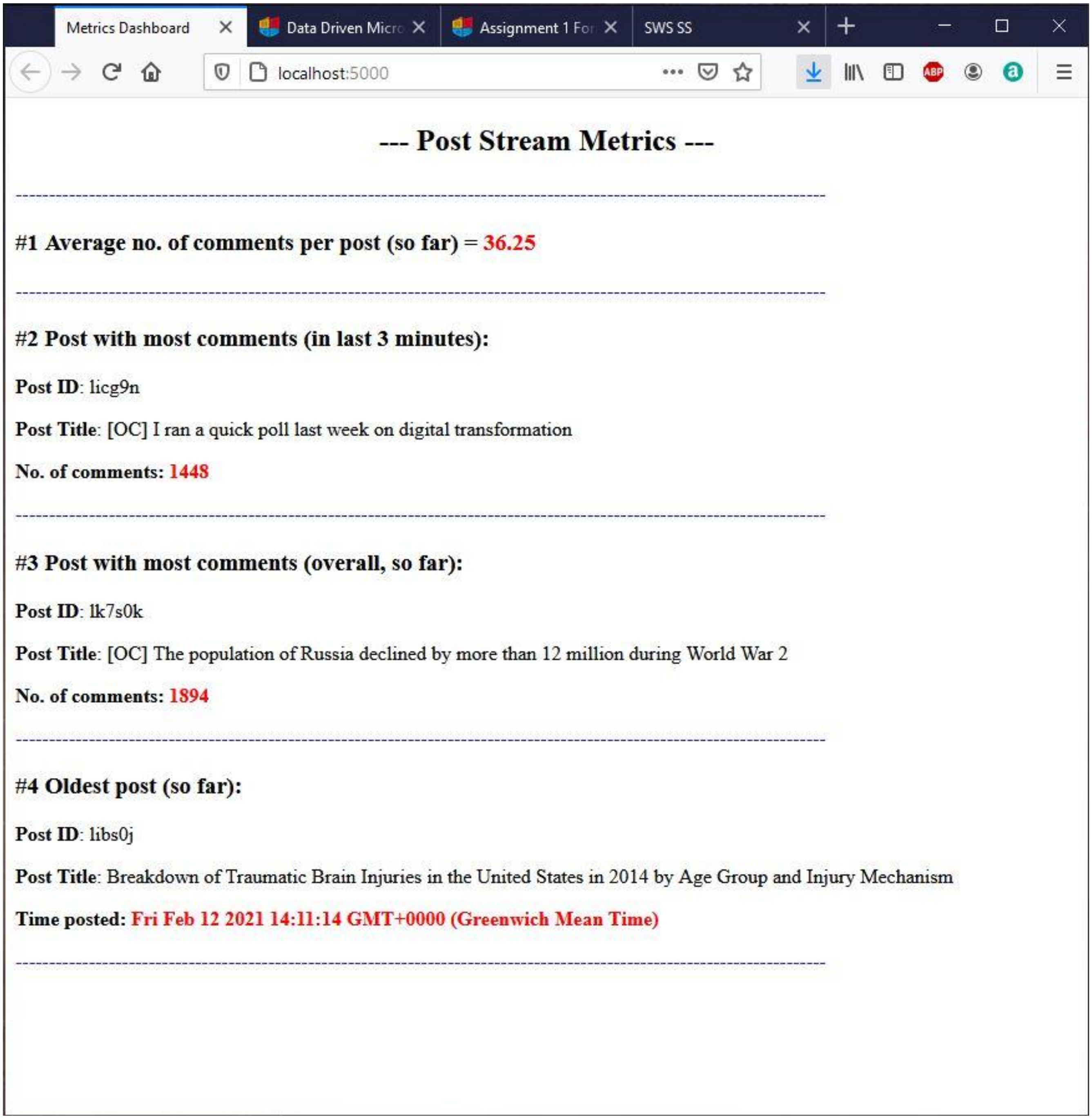
Do you have your Dockerfiles and other microservice files (python scripts, requirements.txt, etc) in separate folders?	YES 3 services, 3 folders
Do you have a functioning Docker Compose file?	YES
Are you streaming data, simulating real-time throughout?	YES CSV data is continuously streamed from GRPC server to client once client calls remote procedure in server.
Do you have 4 pieces of analytics calculated, including a window-on-data (e.g. rolling 3-minute window)?	YES 1. Average no. of comments per post (so far) 2. Post with most comments (in last 3 minutes) * 3. Post with most comments (overall, so far) 4. Oldest post (so far) *Rolling window metric
Are you displaying live analytics on a web page?	YES Available on **localhost:5000** once **docker-compose up** is run from project folder.
Any other comments, e.g. anything you want to highlight that could otherwise be missed by the marker?	Please see the included README.txt file for a detailed explanation of operation.

N.B. The **'r_dataisbeautiful_posts.csv'** dataset file is **NOT** included with the ZIP and needs to be placed in the **'grpc_server'** folder for the application to work.

Appendix – Screenshot(s) of your application running (e.g. Log output as it starts up, any log output as it runs, any web pages, etc.)

See next page…

The web page displaying the metrics….



--- Post Stream Metrics ---

#1 Average no. of comments per post (so far) = 36.25

#2 Post with most comments (in last 3 minutes):

Post ID: licg9n

Post Title: [OC] I ran a quick poll last week on digital transformation

No. of comments: 1448

#3 Post with most comments (overall, so far):

Post ID: lk7s0k

Post Title: [OC] The population of Russia declined by more than 12 million during World War 2

No. of comments: 1894

#4 Oldest post (so far):

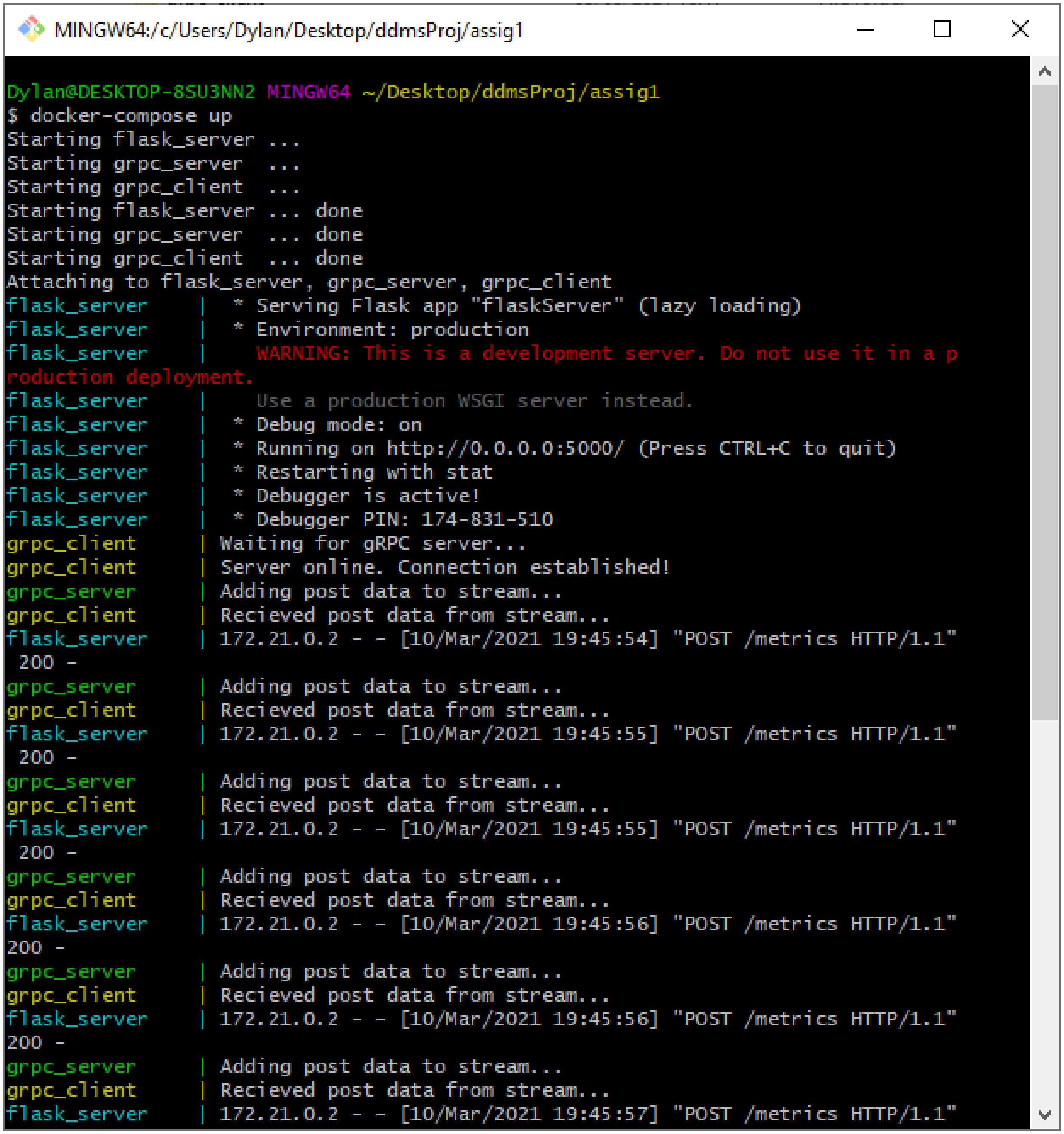
Post ID: libs0j

Post Title: Breakdown of Traumatic Brain Injuries in the United States in 2014 by Age Group and Injury Mechanism

Time posted: Fri Feb 12 2021 14:11:14 GMT+0000 (Greenwich Mean Time)

More on next page….

Terminal output after running ‘docker-compose up’ from project folder…



```
Dylan@DESKTOP-8SU3NN2 MINGW64 ~/Desktop/ddmsProj/assig1
$ docker-compose up
Starting flask_server ...
Starting grpc_server  ...
Starting grpc_client  ...
Starting flask_server ... done
Starting grpc_server  ... done
Starting grpc_client  ... done
Attaching to flask_server, grpc_server, grpc_client
flask_server    |  * Serving Flask app "flaskServer" (lazy loading)
flask_server    |  * Environment: production
flask_server    |    WARNING: This is a development server. Do not use it in a p
roduction deployment.
flask_server    |    Use a production WSGI server instead.
flask_server    |  * Debug mode: on
flask_server    |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
flask_server    |  * Restarting with stat
flask_server    |  * Debugger is active!
flask_server    |  * Debugger PIN: 174-831-510
grpc_client     | Waiting for gRPC server...
grpc_client     | Server online. Connection established!
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:54] "POST /metrics HTTP/1.1"
 200 -
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:55] "POST /metrics HTTP/1.1"
 200 -
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:55] "POST /metrics HTTP/1.1"
 200 -
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:56] "POST /metrics HTTP/1.1"
200 -
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:56] "POST /metrics HTTP/1.1"
200 -
grpc_server     | Adding post data to stream...
grpc_client     | Recieved post data from stream...
flask_server    | 172.21.0.2 - - [10/Mar/2021 19:45:57] "POST /metrics HTTP/1.1"
```

Note: Project was developed on my Linux Mint laptop but I am running it here on Windows using a GIT bash terminal.