

Lab 4.1 Stretch - ClarityNLP

David Strube

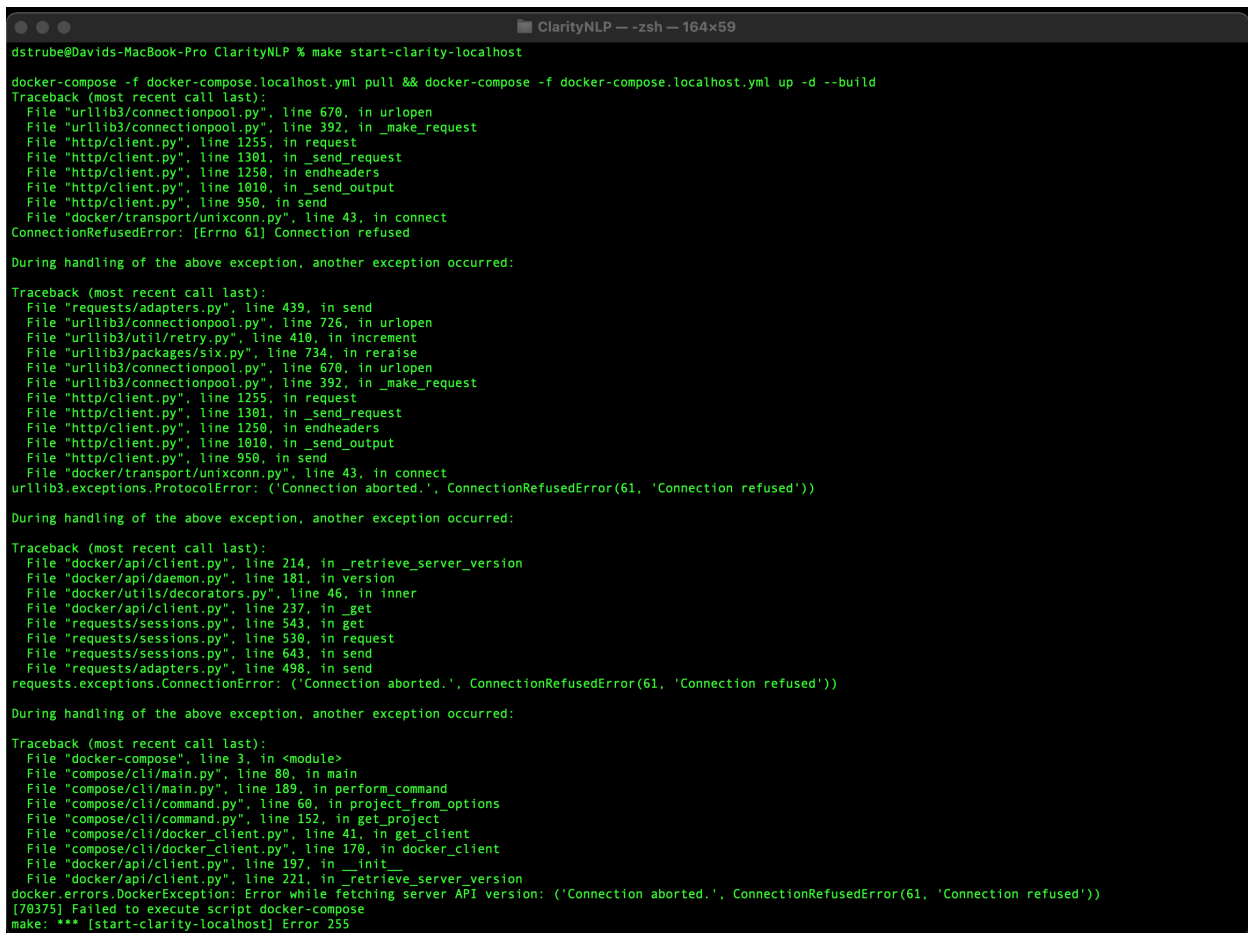
dstrube3@gatech.edu

Abstract — This is the Lab 4.1 Stretch assignment for ClarityNLP for CS 6440 - Intro to Health Informatics. This document includes screenshots for Exercise #2 - NLPQL Phenotype Walkthrough. It also has a link to a recording of a demonstration of different features of ClarityNLP as described in Exercise #3 - ClarityNLP Demonstration.

1 EXERCISE #2 - NLPQL PHENOTYPE WALKTHROUGH

Below are screenshots of following the instructions in the documentation to walk through an NLPQL Phenotype (ClarityNLP, 2018).

When trying to execute the command `make start-clarity-localhost`, if docker is not running, you'll get errors like this:

A terminal window titled 'ClarityNLP -- zsh -- 164x59' showing the execution of 'make start-clarity-localhost'. The command runs 'docker-compose -f docker-compose.localhost.yml pull' followed by 'docker-compose -f docker-compose.localhost.yml up -d --build'. The output shows a series of connection errors. The first error is a 'urllib3.exceptions.ConnectionError' with a 'ConnectionRefusedError(61, 'Connection refused')' message. This is followed by a 'requests.exceptions.ConnectionError' with a 'ConnectionRefusedError(61, 'Connection refused')' message. The final error is a 'docker.errors.DockerException' with a 'ConnectionRefusedError(61, 'Connection refused')' message. The terminal ends with 'make: *** [start-clarity-localhost] Error 255'.

```
dstrube@Davids-MacBook-Pro ClarityNLP % make start-clarity-localhost
docker-compose -f docker-compose.localhost.yml pull && docker-compose -f docker-compose.localhost.yml up -d --build
Traceback (most recent call last):
  File "urllib3/connectionpool.py", line 670, in urlopen
  File "urllib3/connectionpool.py", line 392, in _make_request
  File "http/client.py", line 1255, in request
  File "http/client.py", line 1381, in _send_request
  File "http/client.py", line 1250, in endheaders
  File "http/client.py", line 1010, in _send_output
  File "http/client.py", line 958, in send
  File "docker/transport/unixconn.py", line 43, in connect
ConnectionRefusedError: [Errno 61] Connection refused

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "requests/adapters.py", line 439, in send
  File "urllib3/connectionpool.py", line 726, in urlopen
  File "urllib3/util/retry.py", line 410, in increment
  File "urllib3/packages/six.py", line 734, in reraise
  File "urllib3/connectionpool.py", line 670, in urlopen
  File "urllib3/connectionpool.py", line 392, in _make_request
  File "http/client.py", line 1255, in request
  File "http/client.py", line 1381, in _send_request
  File "http/client.py", line 1250, in endheaders
  File "http/client.py", line 1010, in _send_output
  File "http/client.py", line 958, in send
  File "docker/transport/unixconn.py", line 43, in connect
urllib3.exceptions.ProtocolError: ('Connection aborted.', ConnectionRefusedError(61, 'Connection refused'))

During handling of the above exception, another exception occurred:

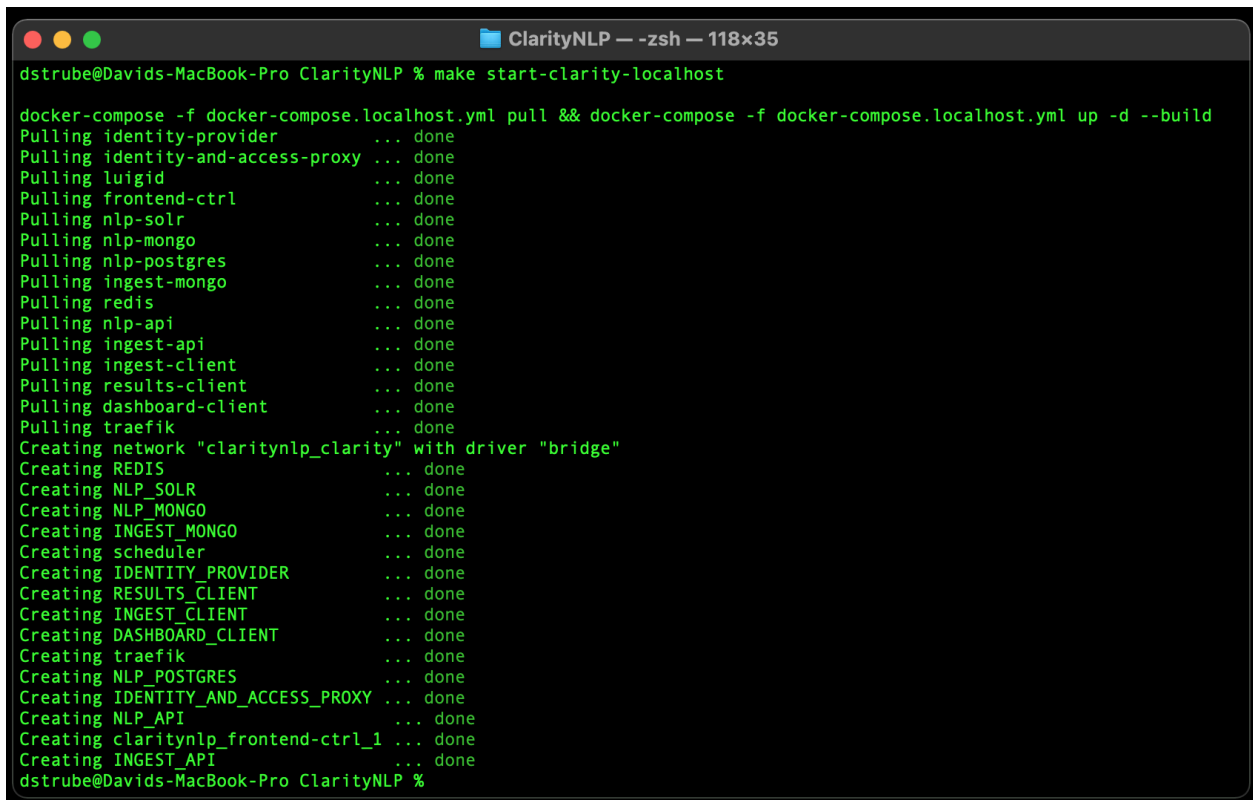
Traceback (most recent call last):
  File "docker/api/client.py", line 214, in _retrieve_server_version
  File "docker/api/daemon.py", line 181, in version
  File "docker/utils/decorators.py", line 46, in inner
  File "docker/api/client.py", line 237, in _get
  File "requests/sessions.py", line 543, in get
  File "requests/sessions.py", line 530, in request
  File "requests/sessions.py", line 643, in send
  File "requests/adapters.py", line 498, in send
requests.exceptions.ConnectionError: ('Connection aborted.', ConnectionRefusedError(61, 'Connection refused'))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "docker-compose", line 3, in <module>
  File "compose/cli/main.py", line 88, in main
  File "compose/cli/main.py", line 189, in perform_command
  File "compose/cli/command.py", line 60, in project_from_options
  File "compose/cli/command.py", line 152, in get_project
  File "compose/cli/docker_client.py", line 41, in get_client
  File "compose/cli/docker_client.py", line 170, in docker_client
  File "docker/api/client.py", line 197, in __init__
  File "docker/api/client.py", line 221, in _retrieve_server_version
docker.errors.DockerException: Error while fetching server API version: ('Connection aborted.', ConnectionRefusedError(61, 'Connection refused'))
[70375] Failed to execute script docker-compose
make: *** [start-clarity-localhost] Error 255
```

Figure 1—Error start up if docker is not running.

When executing the command `make start-clarity-localhost`, if docker is running, you'll get a response like this:



```
dstrube@Davids-MacBook-Pro ClarityNLP % make start-clarity-localhost

docker-compose -f docker-compose.localhost.yml pull && docker-compose -f docker-compose.localhost.yml up -d --build
Pulling identity-provider      ... done
Pulling identity-and-access-proxy ... done
Pulling luigid                ... done
Pulling frontend-ctrl        ... done
Pulling nlp-solr              ... done
Pulling nlp-mongo             ... done
Pulling nlp-postgres          ... done
Pulling ingest-mongo          ... done
Pulling redis                 ... done
Pulling nlp-api               ... done
Pulling ingest-api            ... done
Pulling ingest-client          ... done
Pulling results-client         ... done
Pulling dashboard-client       ... done
Pulling traefik               ... done
Creating network "claritynlp_clarity" with driver "bridge"
Creating REDIS                 ... done
Creating NLP_SOLR              ... done
Creating NLP_MONGO             ... done
Creating INGEST_MONGO          ... done
Creating scheduler             ... done
Creating IDENTITY_PROVIDER      ... done
Creating RESULTS_CLIENT         ... done
Creating INGEST_CLIENT          ... done
Creating DASHBOARD_CLIENT       ... done
Creating traefik               ... done
Creating NLP_POSTGRES           ... done
Creating IDENTITY_AND_ACCESS_PROXY ... done
Creating NLP_API               ... done
Creating claritynlp_frontend-ctrl_1 ... done
Creating INGEST_API            ... done
dstrube@Davids-MacBook-Pro ClarityNLP %
```

Figure 2—Response if docker is running.

When opening Docker with ClarityNLP running, it will look like this. Note the options you get when you mouse over most items are: *CLI* (command line interface), *STOP*, *RESTART*, and *DELETE*. For the item *traefik*, the options also include *OPEN IN BROWSER*. For the root ClarityNLP item, the options also include *Open in Visual Studio Code*. (This option may or may not be available if you don't have Visual Studio Code installed.)

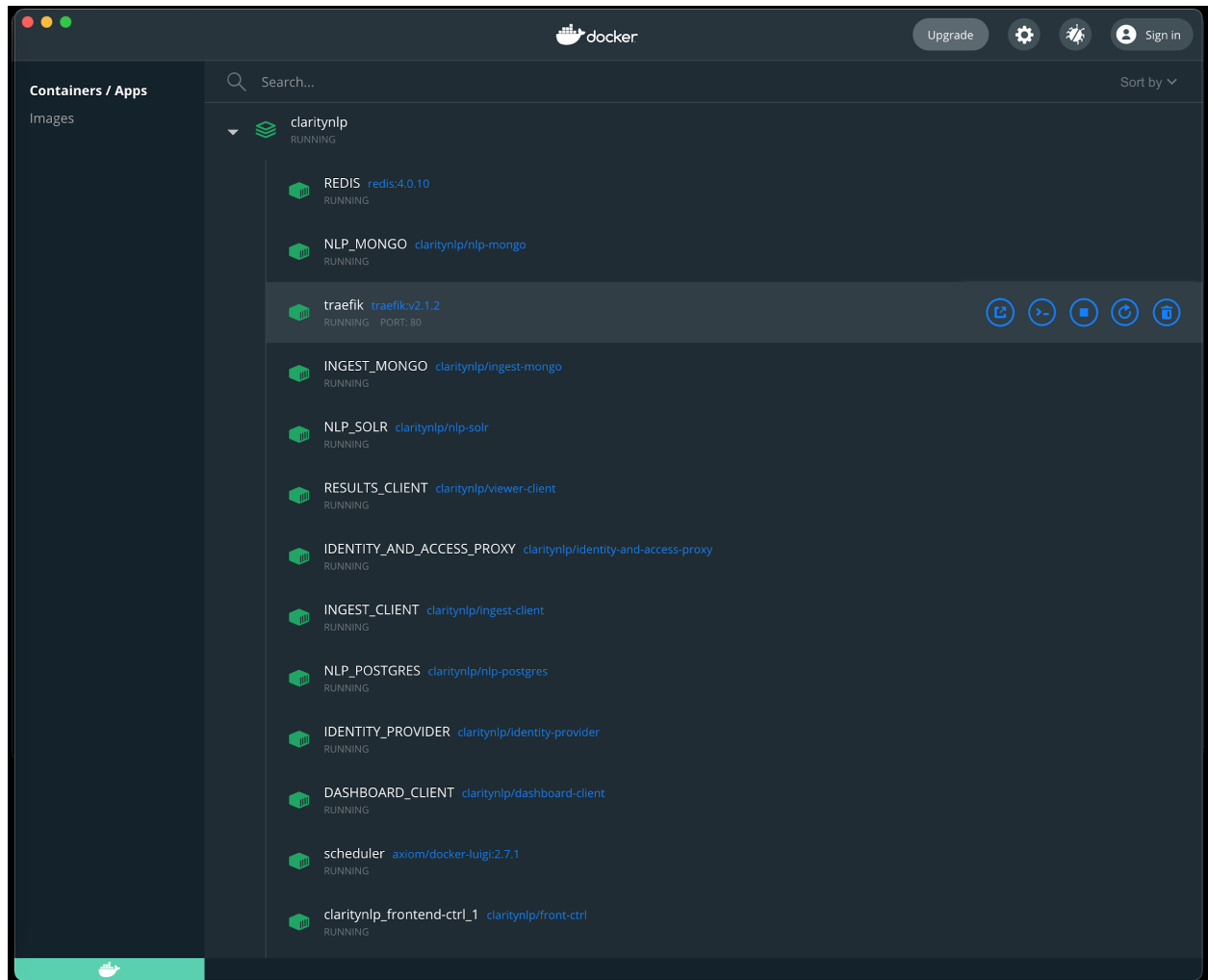


Figure 3—ClarityNLP in Docker.

Opening ClarityNLP in Visual Studio Code looks like this. Note that the file `PatientTemperatures.nlpql` can be edited so that the termset can be renamed, like it is in this screenshot to `TemperatureTerms_dstrube3`:

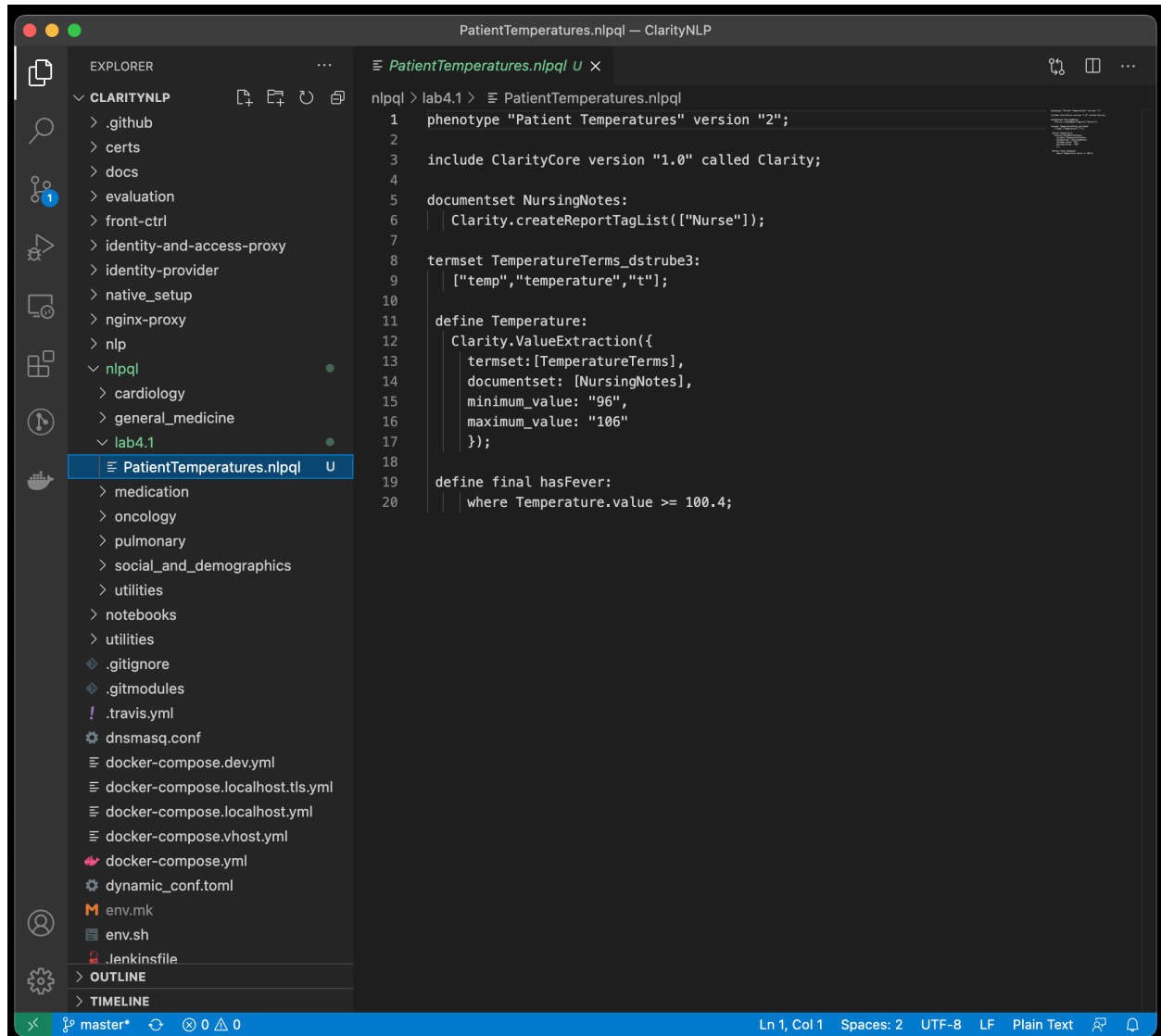


Figure 4—ClarityNLP in Visual Studio Code.

At some point, you may be prompted to install extensions for Python and Pylint in VS Code. This is probably a good idea and will look like this when it's done:

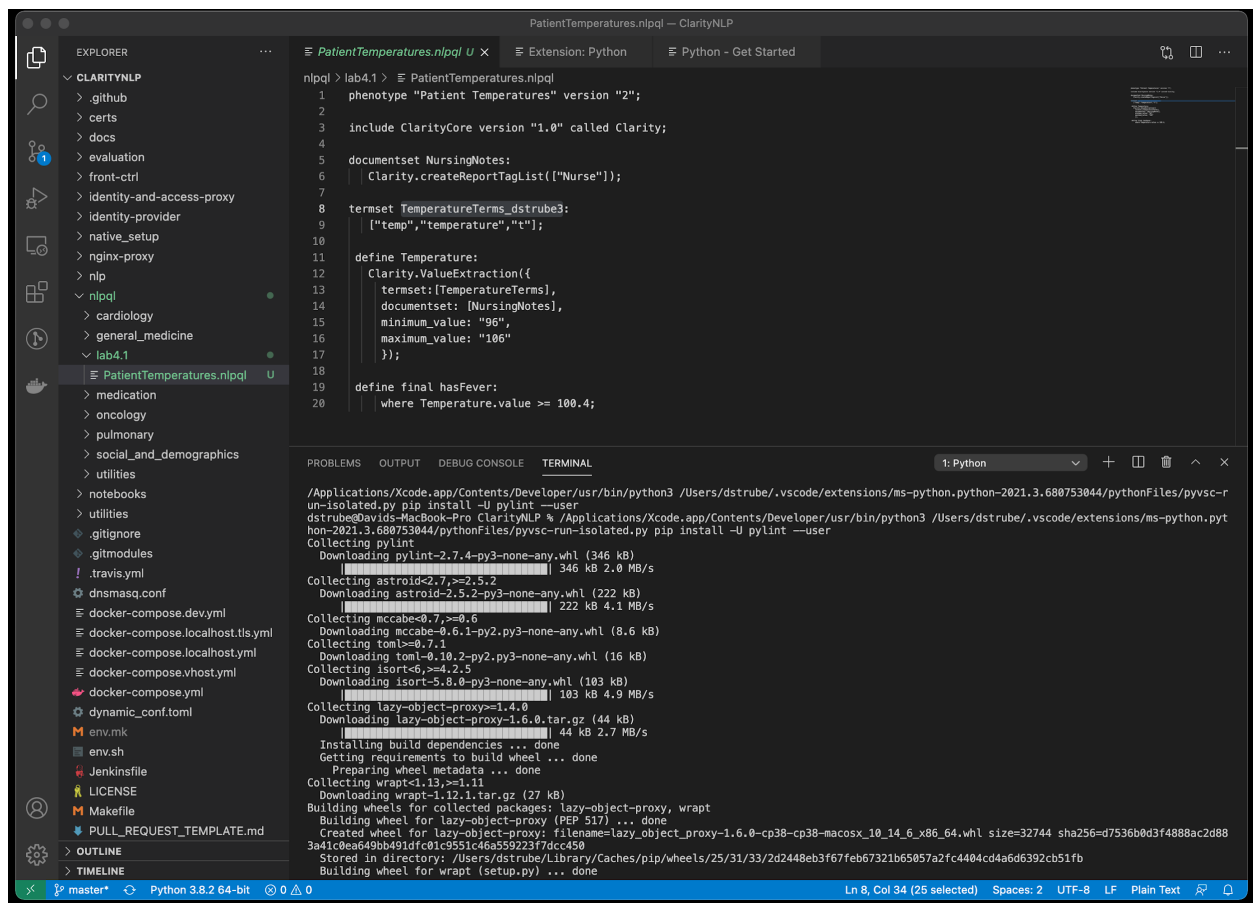
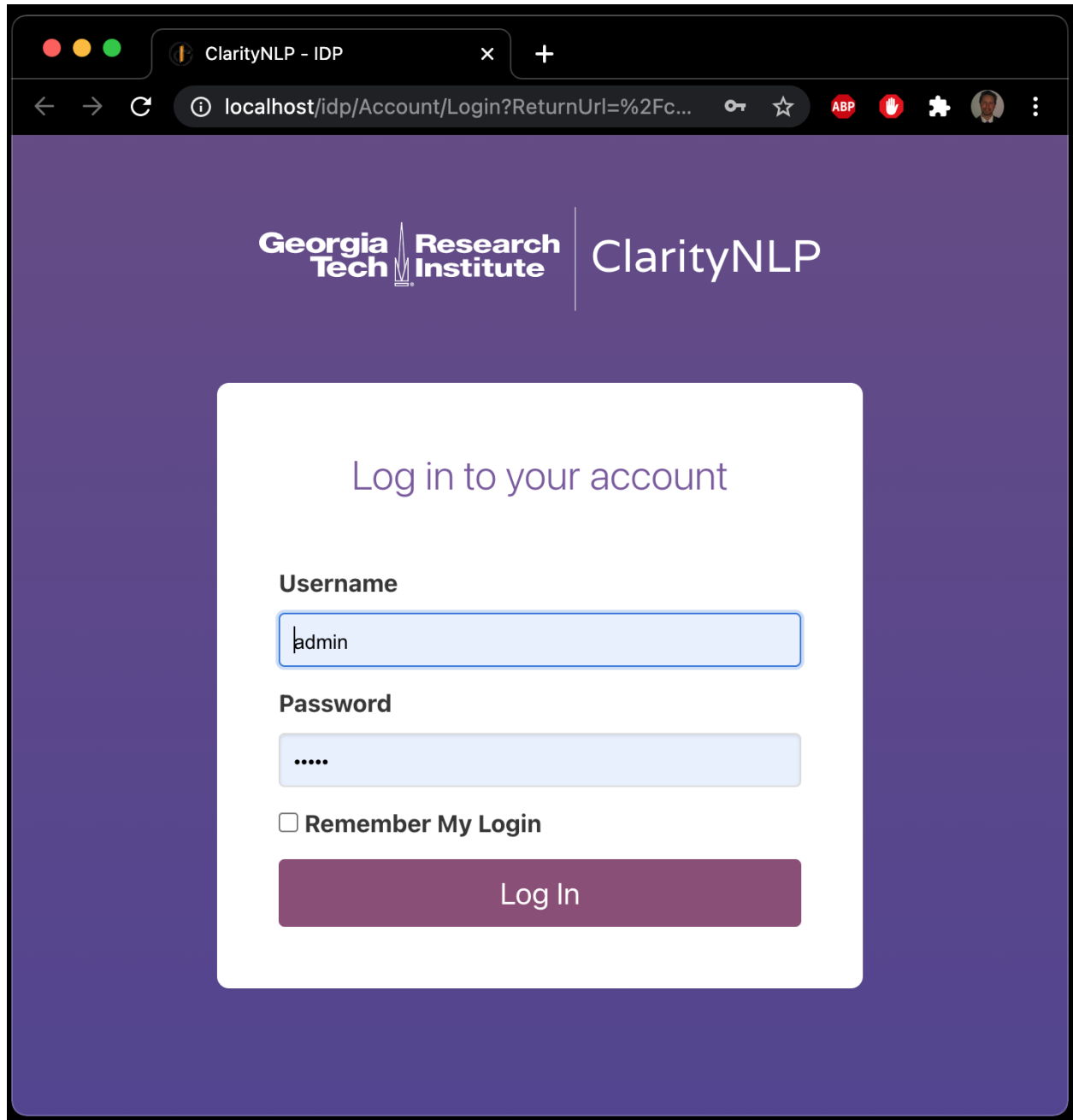


Figure 5—Python in Visual Studio Code.

Once Docker is running ClarityNLP, you can access the ClarityNLP Dashboard at this URL: <http://localhost/dashboard>. You'll get a login screen that looks like this. The username is admin and the password is admin.



The screenshot shows a web browser window with the title "ClarityNLP - IDP". The address bar displays the URL "localhost/idp/Account/Login?ReturnUrl=%2Fc...". The page features the "Georgia Tech Research Institute" logo and the "ClarityNLP" text. A central white box contains the login form with the heading "Log in to your account". The form includes a "Username" field with "admin" entered, a "Password" field with masked characters ".....", a "Remember My Login" checkbox, and a "Log In" button.

Georgia Tech Research Institute | ClarityNLP

Log in to your account

Username

admin

Password

.....

☐ Remember My Login

Log In

The rest of the interface will be explored in the next section.

2 EXERCISE #3 - CLARITYNLP DEMONSTRATION

My demonstration utilizing the different features of ClarityNLP using a local instance can be found here: https://youtu.be/_qUBdu5ddF0.

3 REFERENCES

1. ClarityNLP. (2018). Local Machine Setup With Docker - ClarityNLP documentation. <https://claritynlp.readthedocs.io/en/latest/setup/local-docker.html>. Accessed April 4, 2021.