

# Assignment 2 Part-2

Productionizing the Pipeline-Team7

## Introduction

Duration: 2:00

**The goal of this assignment is to create APIs which can**

- Scrape data from a website
- Recognize entities which can be
- Anonymize data through Masking and Anonymization
- De-anonymize fields that can be de-anonymized

Additionally, we will create a **\*\*Streamlit\*\*** interface where we will test the APIs and display the results.

## APIs created using FastAPI

Duration: 2:00

**Scraping:** This API scrapes data from the website: <https://seekingalpha.com/earnings/earnings-call-transcripts> using beautifulsoup bs4 library. The data was scraped from each of the links provided on the web page and stored in a S3 bucket in txt file format.

**Entity Recognition:** Here we have read the txt files from the S3 bucket and called the Amazon Comprehend Service to detect the PII information. This identified entities are stored in a S3 bucket

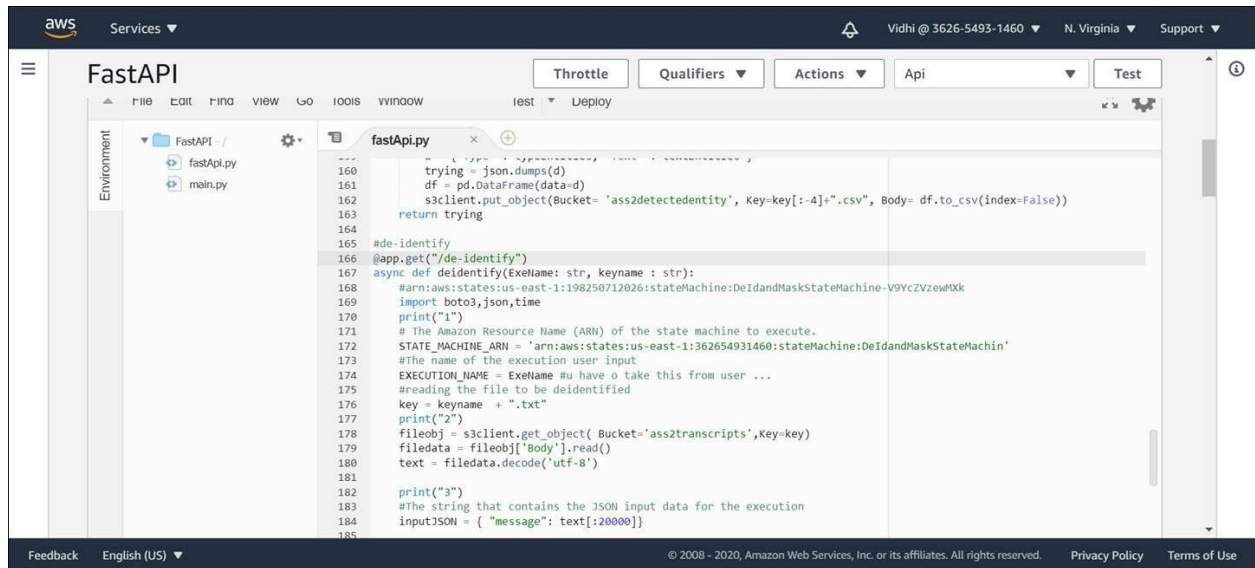
**De-identify:** In this Api the de-identification Step function was leveraged to get the anonymized message.

**Masking:** In this Api the de-identification Step function was leveraged to get the anonymized message.

**Re-identify:** The Api re-identifies the entities from the mentioned text file with the help of the given hash key.

# Assignment 2 Part-2

## Productionizing the Pipeline-Team7



## API Gateway

Duration: 2:00

- The FastAPI implementation is packaged using Magnum and deployed on AWS Lambda. This is done by zipping up the main.py file of fastAPI along with its python dependencies.
- The handler is set to fastApi.handler(wrapped by magnum) and tested and deployed on AWS Lambda. Note: In order to check the code on lambda, the dependencies are added as layers above the main fastApi file.
- A RestApi based API is created with complete control over the request and response of API management capabilities. It can be created with the first method 'ANY' or 'GET' with integration of Lambda functions created in the above step. Then the API is deployed which hits the fastAPI lambda and can be accessed using the link generated.

# Assignment 2 Part-2

Productionizing the Pipeline-Team7

## Streamlit Application

Duration: 2:00

Streamlit python was used to create the front-end for our application.

Below are all the files and their descriptions in the Streamlit folder:

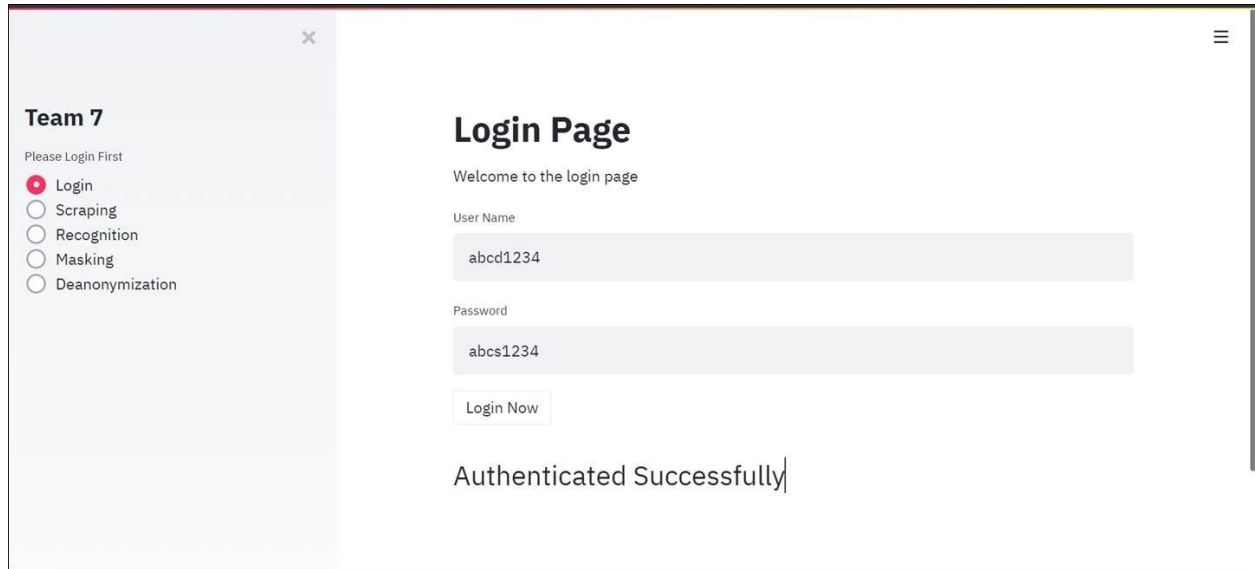
- main.py- the main page containing the sidebar and page section choose from
- login.py- contains the view to login into the app using AWS Cognito and DynamoDB. We will have to enter the username and the password to login and continue with running all the tasks/API
- scraping.py- contains the view and API to scrape data from a particular URL. We will enter the URL as the input and then all the scraped data for multiple web pages will be stored in an S3 bucket as txt files
- recognition.py- contains a view and API to recognize each entity from the article/text. PII information is detected by the API using AWS Comprehend Service from the text files
- masking.py- contains a view and API to mask entities from the text. The recognized entities will be displayed as masked data in the page after the API is called
- deanonymization.py- contains a view and API to de-anonymize entities from the anonymized text. We use a hash algorithm to deanonymize the anonymize data in a previous API and display the data in this page

# Assignment 2 Part-2

## Productionizing the Pipeline-Team7

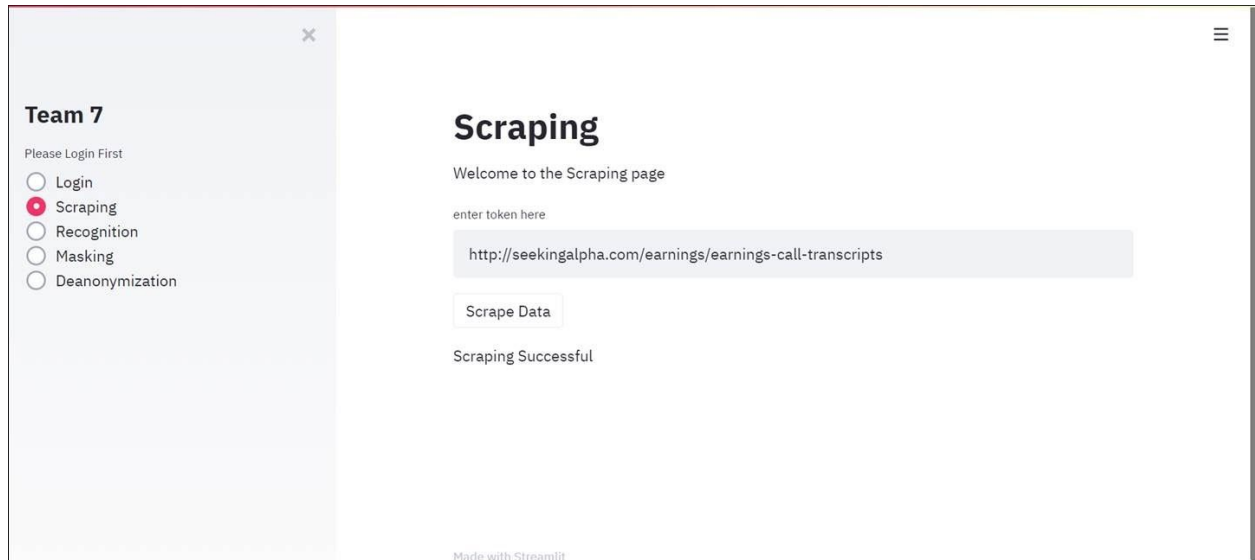
Post authentication, each page will check if the user is logged in before actually calling the API. This will ensure that only if the user is logged in, will the data be displayed in the respective pages.

Enter the correct login credentials-



The screenshot shows a web application interface for 'Team 7'. On the left is a sidebar with a close button (X) and a menu icon (≡). The sidebar contains the text 'Team 7' and 'Please Login First', followed by a list of radio buttons: 'Login' (selected), 'Scraping', 'Recognition', 'Masking', and 'Deanonymization'. The main content area is titled 'Login Page' and includes a welcome message 'Welcome to the login page'. It features two input fields: 'User Name' with the value 'abcd1234' and 'Password' with the value 'abcs1234'. Below these fields is a 'Login Now' button. At the bottom of the main area, the text 'Authenticated Successfully' is displayed.

Enter the correct url to be scraped-



The screenshot shows the same web application interface, but now on the 'Scraping' page. The sidebar remains the same, with 'Scraping' now selected in the radio button list. The main content area is titled 'Scraping' and includes a welcome message 'Welcome to the Scraping page'. It features an input field labeled 'enter token here' with the value 'http://seekingalpha.com/earnings/earnings-call-transcripts'. Below this field is a 'Scrape Data' button. At the bottom of the main area, the text 'Scraping Successful' is displayed. A footer at the very bottom of the page reads 'Made with Streamlit'.

# Assignment 2 Part-2

## Productionizing the Pipeline-Team7

No inputs needed in this page. Directly press the button-

Team 7

Please Login First

☐ Login

☐ Scraping

☒ Recognition

☐ Masking

☐ Deanonimization

Entity Identification

Welcome to the Entity Recognition page

Identify Entities

Comprehension successful

```
{ "Type": ["ORGANIZATION", "ORGANIZATION", "ORGANIZATION", "EVENT", "DATE", "DATE", "PERSON", "PERSON", "PERSON", "PERSON", "PERSON", "PERSON", "PERSON", "PERSON", "ORGANIZATION", "ORGANIZATION", "DATE", "LOCATION", "DATE", "QUANTITY", "QUANTITY", "PERSON", "QUANTITY", "OTHER", "QUANTITY", "QUANTITY", "DATE", "DATE", "QUANTITY", "OTHER", "QUANTITY", "QUANTITY", "DATE", "DATE", "OTHER", "QUANTITY", "ORGANIZATION", "QUANTITY", "DATE", "QUANTITY", "DATE", "QUANTITY", "QUANTITY", "DATE", "QUANTITY", "DATE", "QUANTITY", "OTHER", "DATE", "DATE", "DATE", "DATE", "DATE", "ORGANIZATION", "QUANTITY", "QUANTITY", "QUANTITY", "DATE", "QUANTITY", "DATE", "ORGANIZATION", "DATE", "QUANTITY", "DATE", "DATE", "DATE", "DATE", "QUANTITY", "QUANTITY", "OTHER", "PERSON", "DATE", "QUANTITY", "DATE", "DATE", "OTHER", "DATE", "PERSON", "DATE", "LOCATION", "LOCATION", "PERSON", "LOCATION", "LOCATION", "QUANTITY", "QUANTITY", "DATE", "QUANTITY", "QUANTITY", "ORGANIZATION", "ORGANIZATION", "DATE", "QUANTITY"], "Text": ["Verde Agritech PLC", "OTCQB", "AMHPF", "Q3 2020", "November 24, 2020", "11:30 AM EST"] }
```

Team 7

Please Login First

☐ Login

☐ Scraping

☐ Recognition

☐ Masking

☐ Anonimization

☒ Deanonimization

De-Anonymization

Welcome to the De-anonymization page

enter Hash here

5978578d68e764a49415694396ca01e3fcc2b4616bd656731dd2c32e5b9cdfdc

enter filename here

4391288-burlington-stores-inc-burl-ceo-michael-osullivan-on-q3-2020-results-earnings-ca

De-Anonymize Data

Deanonimization successful

"Burlington Stores, Inc. (NYSE:BURL) Q3 2020 Results Conference Call Novemer 24, 2020 8:30 AM ET Company Participants David Glick - SVP, IR and Treasurer Michael O'Sullivan - CEO John Crimmins - CFO Conference Call Participants Matthew Boss - JP Morgan Ike Boruchow - Wells Fargo John Kernan - Cowen Lorraine Hutchinson - Bank of America Kimerly Greenerger - Morgan Stanley Operator Ladies and gentlemen, thank you for standing y. And welcome to the Burlington Stores Incorporated Third Quarter 2020 Earnings Wecast Call. At this time, all participants are in a listen-only mode. [Operator Instructions] Please e advised that today's conference is eing recorded. [Operator Instructions] I would now like to hand the conference over to your speaker today, David Glick, Senior Vice President of Investor Relations and Treasurer. Please go ahead. David Glick Thank you, operator, and good morning, everyone. We appreciate

# Assignment 2 Part-2

Productionizing the Pipeline-Team7