fags = """About the Program

What is the course fee for Data Science Mentorship Program (DSMP 2023)

The course follows a monthly subscription model where you have to make monthly payments of What is the total duration of the course?

The total duration of the course is 7 months. So the total course fee becomes 799\*7 = Rs 5 What is the syllabus of the mentorship program?

We will be covering the following modules:

Python Fundamentals

Python libraries for Data Science

Data Analysis

SQL for Data Science

Maths for Machine Learning

ML Algorithms

Practical ML

MLOPs

Case studies

You can check the detailed syllabus here - https://learnwith.campusx.in/courses/CampusX-Da Will Deep Learning and NLP be a part of this program?

No, NLP and Deep Learning both are not a part of this program's curriculum.

What if I miss a live session? Will I get a recording of the session?

Yes all our sessions are recorded, so even if you miss a session you can go back and watch Where can I find the class schedule?

Checkout this google sheet to see month by month time table of the course - https://docs.go What is the time duration of all the live sessions?

Roughly, all the sessions last 2 hours.

What is the language spoken by the instructor during the sessions?

Hinglish

How will I be informed about the upcoming class?

You will get a mail from our side before every paid session once you become a paid user.

Can I do this course if I am from a non-tech background?

Yes, absolutely.

I am late, can I join the program in the middle?

Absolutely, you can join the program anytime.

If I join/pay in the middle, will I be able to see all the past lectures?

Yes, once you make the payment you will be able to see all the past content in your dashbowhere do I have to submit the task?

You don't have to submit the task. We will provide you with the solutions, you have to sel Will we do case studies in the program?
Yes.

Where can we contact you?

You can mail us at nitish.campusx@gmail.com

Payment/Registration related questions

Where do we have to make our payments? Your YouTube channel or website?

You have to make all your monthly payments on our website. Here is the link for our websit Can we pay the entire amount of Rs 5600 all at once?

Unfortunately no, the program follows a monthly subscription model.

What is the validity of monthly subscription? Suppose if I pay on 15th Jan, then do I have 15th Feb. The validity period is 30 days from the day you make the payment. So essentially What if I don't like the course after making the payment. What is the refund policy?

You get a 7 days refund period from the day you have made the payment.

I am living outside India and I am not able to make the payment on the website, what shoul You have to contact us by sending a mail at nitish.campusx@gmail.com

Post registration queries

Till when can I view the paid videos on the website?

This one is tricky, so read carefully. You can watch the videos till your subscription is But once the course is over and you have paid us Rs 5600(or 7 installments of Rs 799) you why lifetime validity is not provided?

Because of the low course fee.

```
Where can I reach out in case of a doubt after the session?
You will have to fill a google form provided in your dashboard and our team will contact y
If I join the program late, can I still ask past week doubts?
Yes, just select past week doubt in the doubt clearance google form.
I am living outside India and I am not able to make the payment on the website, what shoul
You have to contact us by sending a mail at nitish.campusx@gmai.com
Certificate and Placement Assistance related queries
What is the criteria to get the certificate?
There are 2 criterias:
You have to pay the entire fee of Rs 5600
You have to attempt all the course assessments.
I am joining late. How can I pay payment of the earlier months?
You will get a link to pay fee of earlier months in your dashboard once you pay for the cu
I have read that Placement assistance is a part of this program. What comes under Placemen
This is to clarify that Placement assistance does not mean Placement guarantee. So we dont
Portfolio Building sessions
Soft skill sessions
Sessions with industry mentors
Discussion on Job hunting strategies
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer()
tokenizer.fit on texts([fags])
len(tokenizer.word index)
 → 282
input sequences = []
for sentence in faqs.split('\n'):
  tokenized_sentence = tokenizer.texts_to_sequences([sentence])[0]
  for i in range(1,len(tokenized_sentence)):
    input sequences.append(tokenized sentence[:i+1])
input_sequences
 <del>→</del> [[93, 1],
      [93, 1, 13],
      [11, 7],
      [11, 7, 1],
       [11, 7, 1, 12],
       [11, 7, 1, 12, 42],
       [11, 7, 1, 12, 42, 15],
       [11, 7, 1, 12, 42, 15, 43],
       [11, 7, 1, 12, 42, 15, 43, 53],
      [11, 7, 1, 12, 42, 15, 43, 53, 68],
       [11, 7, 1, 12, 42, 15, 43, 53, 68, 13],
       [11, 7, 1, 12, 42, 15, 43, 53, 68, 13, 147],
```

[11, 7, 1, 12, 42, 15, 43, 53, 68, 13, 147, 148],

```
[1, 12],
[1, 12, 94],
[1, 12, 94, 5],
[1, 12, 94, 5, 44],
[1, 12, 94, 5, 44, 29],
[1, 12, 94, 5, 44, 29, 95],
[1, 12, 94, 5, 44, 29, 95, 30],
[1, 12, 94, 5, 44, 29, 95, 30, 2],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44, 69],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44, 69, 6],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44, 69, 6, 31],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44, 69, 6, 31, 70],
[1, 12, 94, 5, 44, 29, 95, 30, 2, 8, 4, 26, 44, 69, 6, 31, 70, 45],
[11, 7],
[11, 7, 1],
[11, 7, 1, 71],
[11, 7, 1, 71, 72],
[11, 7, 1, 71, 72, 6],
[11, 7, 1, 71, 72, 6, 1],
[11, 7, 1, 71, 72, 6, 1, 12],
[1, 71],
[1, 71, 72],
[1, 71, 72, 6],
[1, 71, 72, 6, 1],
[1, 71, 72, 6, 1, 12],
[1, 71, 72, 6, 1, 12, 7],
[1, 71, 72, 6, 1, 12, 7, 54],
[1, 71, 72, 6, 1, 12, 7, 54, 73],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149, 70],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149, 70, 54],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149, 70, 54, 31],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149, 70, 54, 31, 55],
[1, 71, 72, 6, 1, 12, 7, 54, 73, 32, 1, 71, 12, 42, 149, 70, 54, 31, 55, 150],
[11, 7],
[11, 7, 1],
```

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
padded_input_sequences = pad_sequences(input_sequences, maxlen = max_len, padding='pre')
```

max\_len = max([len(x) for x in input\_sequences])

```
padded_input_sequences
```

```
0, 93,
 → array([[
                          0, ...,
                     0,
                                              1],
                     0,
                          0, ..., 93, 1,
             0, ...,
                                        11,
             Γ
                     0.
                                   0,
                     0,
                          0, ..., 279, 18, 280],
                0,
                          0, ..., 18, 280, 281],
                0,
                     0,
                          0, ..., 280, 281, 282]], dtype=int32)
             Γ
                0,
                     0,
X = padded_input_sequences[:,:-1]
y = padded input sequences[:,-1]
X.shape
 → (863, 56)
y.shape
 → (863,)
from tensorflow.keras.utils import to_categorical
y = to_categorical(y,num_classes=283)
y.shape
 → (863, 283)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense
model = Sequential()
model.add(Embedding(283, 100, input_length=56))
model.add(LSTM(150, return_sequences=True))  # Set return_sequences to True
model.add(LSTM(150))
model.add(Dense(283, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
model.summary()
    Model: "sequential_4"
                                    Output Shape
      Layer (type)
                                                               Param #
```

\_\_\_\_\_\_

<pre>embedding_4 (Embedding)</pre>	(None, 56, 100)	28300
lstm_8 (LSTM)	(None, 56, 150)	150600
lstm_9 (LSTM)	(None, 150)	180600
dense (Dense)	(None, 283)	42733

\_\_\_\_\_

Total params: 402233 (1.53 MB)
Trainable params: 402233 (1.53 MB)
Non-trainable params: 0 (0.00 Byte)

\_\_\_\_\_

## model.fit(X,y,epochs=100)

```
→ Epoch 1/100
Epoch 2/100
Epoch 3/100
Epoch 4/100
Epoch 5/100
Epoch 6/100
Epoch 7/100
Epoch 8/100
Epoch 9/100
Epoch 10/100
Epoch 11/100
Epoch 12/100
Epoch 13/100
Epoch 14/100
Epoch 15/100
Epoch 16/100
Epoch 17/100
Epoch 18/100
Epoch 19/100
Epoch 20/100
```

```
Epoch 21/100
Epoch 22/100
Epoch 23/100
Epoch 24/100
Epoch 25/100
Epoch 26/100
Epoch 27/100
Epoch 28/100
Epoch 29/100
27/27 [----- local 1 0221 | accuracy
```

## pip install numpy tensorflow

Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-pac Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/d Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.1 Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/loca Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10 Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-pa-Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/di Requirement already satisfied: ml-dtypes~=0.2.0 in /usr/local/lib/python3.10/di Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/d Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-pack Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21. Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-pac Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-pa Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/di Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/pytho Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10 Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/loc Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10 Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local Requirement already satisfied: keras<2.16,>=2.15.0 in /usr/local/lib/python3.10 Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/ Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3. Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/p Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dis Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10 Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/lo-Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dis Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3 Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3. Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/pytho Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pytho

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/
    Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/d
    Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.1
    Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dis
import time
text = "total duration of the course"
for i in range(10):
  # tokenize
  token_text = tokenizer.texts_to_sequences([text])[0] # Access the first element of th
  padded_token_text = pad_sequences([token_text], maxlen=56, padding='pre')
  # predict
  pos = np.argmax(model.predict(padded_token_text))
   for word,index in tokenizer.word_index.items():
      if index == pos:
         text = text + " " + word
         print(text)
         time.sleep(2)
 1/1 [======= ] - Os 33ms/step
    1/1 [======= ] - Os 35ms/step
    1/1 [======= ] - Os 35ms/step
    1/1 [======= ] - Os 35ms/step
    1/1 [======= ] - Os 34ms/step
    1/1 [=======] - Os 37ms/step
    1/1 [======= ] - Os 42ms/step
    1/1 [======= ] - Os 35ms/step
import time
text = "what is the fee"
for i in range(10):
 # tokenize
 token_text = tokenizer.texts_to_sequences([text])[0]
 padded token text = pad sequences([token text], maxlen=56, padding='pre')
 # predict
 pos = np.argmax(model.predict(padded token text))
 for word,index in tokenizer.word_index.items():
  if index == pos:
    text = text + " " + word
    print(text)
    time.sleep(2)
 → 1/1 [============= ] - Os 36ms/step
    1/1 [=======] - Os 66ms/step
    1/1 [======= ] - Os 61ms/step
    1/1 [======= ] - Os 59ms/step
    1/1 [======= ] - Os 64ms/step
    1/1 [======= ] - Os 57ms/step
```

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-page 1.00 in /usr/local/lib/pyth

tokenizer.word\_index

