

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 you  
 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 should I do?  
 You have to contact us by sending a mail at nitish.campusx@gmail.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 current month.  
 I have read that Placement assistance is a part of this program. What comes under Placement Assistance?  
 This is to clarify that Placement assistance does not mean Placement guarantee. So we don't have a placement guarantee.  
 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([faqs])
```

```
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
```

↩

```
[[[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],
...

```

```
max_len = max([len(x) for x in input_sequences])
```

```

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

```

```
padded_input_sequences
```

```
⇒ array([[ 0,  0,  0, ...,  0, 93,  1],
         [ 0,  0,  0, ..., 93,  1, 13],
         [ 0,  0,  0, ...,  0, 11,  7],
         ...,
         [ 0,  0,  0, ..., 279, 18, 280],
         [ 0,  0,  0, ..., 18, 280, 281],
         [ 0,  0,  0, ..., 280, 281, 282]], dtype=int32)
```

```
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"
```

Layer (type)	Output Shape	Param #
=====		

embedding_4 (Embedding)	(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
27/27 [=====] - 11s 241ms/step - loss: 5.4243 - accuracy: 0.0000
Epoch 2/100
27/27 [=====] - 5s 176ms/step - loss: 5.1099 - accuracy: 0.0000
Epoch 3/100
27/27 [=====] - 5s 189ms/step - loss: 5.0349 - accuracy: 0.0000
Epoch 4/100
27/27 [=====] - 6s 218ms/step - loss: 5.0254 - accuracy: 0.0000
Epoch 5/100
27/27 [=====] - 5s 176ms/step - loss: 5.0157 - accuracy: 0.0000
Epoch 6/100
27/27 [=====] - 6s 240ms/step - loss: 5.0064 - accuracy: 0.0000
Epoch 7/100
27/27 [=====] - 5s 181ms/step - loss: 5.0066 - accuracy: 0.0000
Epoch 8/100
27/27 [=====] - 6s 212ms/step - loss: 4.9911 - accuracy: 0.0000
Epoch 9/100
27/27 [=====] - 5s 194ms/step - loss: 4.9550 - accuracy: 0.0000
Epoch 10/100
27/27 [=====] - 5s 200ms/step - loss: 4.8672 - accuracy: 0.0000
Epoch 11/100
27/27 [=====] - 7s 237ms/step - loss: 4.7065 - accuracy: 0.0000
Epoch 12/100
27/27 [=====] - 5s 177ms/step - loss: 4.5133 - accuracy: 0.0000
Epoch 13/100
27/27 [=====] - 6s 238ms/step - loss: 4.3492 - accuracy: 0.0000
Epoch 14/100
27/27 [=====] - 5s 179ms/step - loss: 4.1745 - accuracy: 0.0000
Epoch 15/100
27/27 [=====] - 5s 187ms/step - loss: 3.9668 - accuracy: 0.0000
Epoch 16/100
27/27 [=====] - 6s 224ms/step - loss: 3.7731 - accuracy: 0.0000
Epoch 17/100
27/27 [=====] - 5s 182ms/step - loss: 3.5790 - accuracy: 0.0000
Epoch 18/100
27/27 [=====] - 6s 231ms/step - loss: 3.4063 - accuracy: 0.0000
Epoch 19/100
27/27 [=====] - 5s 176ms/step - loss: 3.2531 - accuracy: 0.0000
Epoch 20/100

```

```

27/27 [=====] - 6s 207ms/step - loss: 3.0940 - accuracy: 0.4500
Epoch 21/100
27/27 [=====] - 6s 201ms/step - loss: 2.9512 - accuracy: 0.4500
Epoch 22/100
27/27 [=====] - 5s 177ms/step - loss: 2.7897 - accuracy: 0.4500
Epoch 23/100
27/27 [=====] - 6s 236ms/step - loss: 2.6693 - accuracy: 0.4500
Epoch 24/100
27/27 [=====] - 5s 177ms/step - loss: 2.5177 - accuracy: 0.4500
Epoch 25/100
27/27 [=====] - 6s 227ms/step - loss: 2.3800 - accuracy: 0.4500
Epoch 26/100
27/27 [=====] - 5s 177ms/step - loss: 2.2704 - accuracy: 0.4500
Epoch 27/100
27/27 [=====] - 5s 177ms/step - loss: 2.1554 - accuracy: 0.4500
Epoch 28/100
27/27 [=====] - 6s 236ms/step - loss: 2.0363 - accuracy: 0.4500
Epoch 29/100
27/27 [=====] - 5s 177ms/step - loss: 1.9221 - accuracy: 0.4500

```

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-packages
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: ml-dtypes~=0.2.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: keras<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages

```

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (2.5)  
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (1.26.15)  
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (2024.7.4)  
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (2.1.5)  
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (0.6.0)  
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (3.2.2)

```
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 the list
    # padding
    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 [=====] - 1s 826ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 35ms/step
```

```
import time
text = "what is the fee"

for i in range(10):
    # tokenize
    token_text = tokenizer.texts_to_sequences([text])[0]
    # padding
    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 [=====] - 0s 36ms/step
1/1 [=====] - 0s 66ms/step
1/1 [=====] - 0s 61ms/step
1/1 [=====] - 0s 59ms/step
1/1 [=====] - 0s 64ms/step
1/1 [=====] - 0s 57ms/step
```

```
1/1 [=====] - 0s 55ms/step
1/1 [=====] - 0s 69ms/step
1/1 [=====] - 0s 51ms/step
1/1 [=====] - 0s 52ms/step
```

```
tokenizer.word_index
```

```
↔ {}
```