chatbot.py 1 | new.py | intents.json

chatbot > new.py > ...

```python
import random
import json
import pickle
import numpy as np
import tensorflow as tf

import nltk
from nltk.stem import WordNetLemmatiz

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('wordnet')
```

new.py

chatbot > new.py > ...

```python
# Load intents file
intents = json.loads(open(r'C:\Users\moh

words = []
classes = []
documents = []
ignoreLetters = ['?', '!', ',', '.']

# Tokenize each word in the patterns and
for intent in intents['intents']:
    for pattern in intent['patterns']:
        wordList = nltk.word_tokenize(pa
```

new.py

chatbot > new.py > ...

```python
training = []
outputEmpty = [0] * len(classes)

# Create training data
for document in documents:
    bag = []
    wordPatterns = document[0]
    wordPatterns = [lemmatizer.lemmati
    for word in words:
        bag.append(1) if word in wordPa

    outputRow = list(outputEmpty)
    outputRow[classes.index(document[1
```

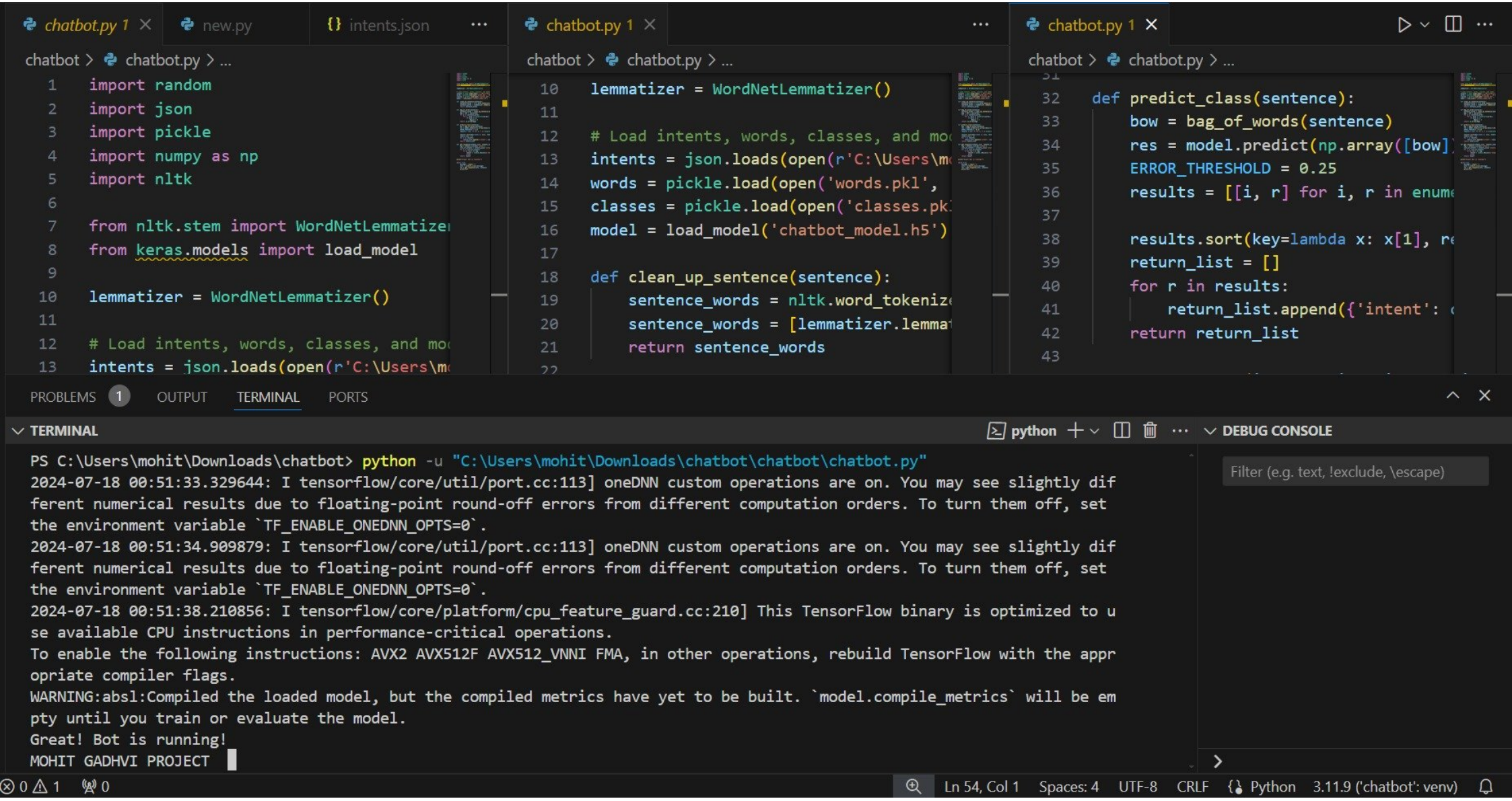
PROBLEMS 1 | OUTPUT | TERMINAL | PORTS

TERMINAL

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 1.0000 - loss: 0.0023
Epoch 197/200
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 1.0000 - loss: 0.0055
Epoch 198/200
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 1.0000 - loss: 0.0025
Epoch 199/200
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 1.0000 - loss: 0.0062
Epoch 200/200
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 1.0000 - loss: 0.0034
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format
 is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving
.save_model(model, 'my_model.keras')`.
Executed
```

DEBUG CONSOLE

Filter (e.g. text, !exclude, \escape)


chatbot.py 1
new.py
intents.json
chatbot > chatbot.py > ...
import random
import json
import pickle
import numpy as np
import nltk
from nltk.stem import WordNetLemmatizer
from keras.models import load_model
lemmatizer = WordNetLemmatizer()
# Load intents, words, classes, and mo
intents = json.loads(open(r'C:\Users\m
words = pickle.load(open('words.pkl',
classes = pickle.load(open('classes.pk
model = load_model('chatbot_model.h5')
def clean_up_sentence(sentence):
sentence_words = nltk.word_tokeniz
sentence_words = [lemmatizer.lemma
return sentence_words
def predict_class(sentence):
bow = bag_of_words(sentence)
res = model.predict(np.array([bow]
ERROR_THRESHOLD = 0.25
results = [[i, r] for i, r in enume
results.sort(key=lambda x: x[1], re
return_list = []
for r in results:
return_list.append({'intent':
return return_list
PROBLEMS 1
OUTPUT
TERMINAL
PORTS
TERMINAL
python
DEBUG CONSOLE
PS C:\Users\mohit\Downloads\chatbot> python -u "C:\Users\mohit\Downloads\chatbot\chatbot\chatbot.py"
2024-07-18 00:51:33.329644: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly dif
ferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set
the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-07-18 00:51:34.909879: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly dif
ferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set
the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-07-18 00:51:38.210856: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to u
se available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appr
opriate compiler flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be em
pty until you train or evaluate the model.
Great! Bot is running!
MOHIT GADHVI PROJECT
Filter (e.g. text, !exclude, \escape)
Ln 54, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.11.9 ('chatbot': venv)

chatbot.py 1 | new.py | intents.json

chatbot > chatbot.py > ...

```python
import random
import json
import pickle
import numpy as np
import nltk

from nltk.stem import WordNetLemmatizer
from keras.models import load_model

lemmatizer = WordNetLemmatizer()
```

Output (Ctrl+Shift+U)

chatbot.py 1

chatbot > chatbot.py > ...

```python
lemmatizer = WordNetLemmatizer()

# Load intents, words, classes, and mo
intents = json.loads(open(r'C:\Users\m
words = pickle.load(open('words.pkl',
classes = pickle.load(open('classes.pk
model = load_model('chatbot_model.h5')

def clean_up_sentence(sentence):
    sentence_words = nltk.word_tokeniz
    sentence_words = [lemmatizer.lemma
```

PROBLEMS 1 | OUTPUT | TERMINAL | PORTS

TERMINAL

```
pty until you train or evaluate the model.
Great! Bot is running!
Hi!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
Good to see you again
How are you?
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Hello!
Hello!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Greetings! How can I help?
What is your name?
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
You can call me ChatBot.
```

0 ⚠ 1 | 0 | Ln 54,