# This files contains your custom actions which can be used to run

# custom Python code.

#

# See this guide on how to implement these action:

# https://rasa.com/docs/rasa/core/actions/#custom-actions/

# This is a simple example for a custom action which utters "Hello World!"

# from typing import Any, Text, Dict, List

#

# from rasa\_sdk import Action, Tracker

# from rasa\_sdk.executor import CollectingDispatcher

#

#

# class ActionHelloWorld(Action):

#

# def name(self) -> Text:

# return "action\_hello\_world"

#

# def run(self, dispatcher: CollectingDispatcher,

# tracker: Tracker,

# domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

#

# dispatcher.utter\_message(text="Hello World!")

#

# return []

from typing import Any, Text, Dict, List

from rasa\_sdk import Action, Tracker

from rasa\_sdk.executor import CollectingDispatcher

from rasa\_sdk.forms import FormAction

import json

class ActionHelloWorld(FormAction):

def name(self) -> Text:

return "predict\_form"

@staticmethod

def required\_slots(tracker: Tracker) -> List[Text]:

print("required\_slots(tracker:Tracker)")

return ["name", "caste", "rank", "course"]

def submit(self, dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> List[Dict]:

dispatcher.utter\_message(template="utter\_submit")

return []

# validation for caste

@staticmethod

def caste\_db() -> List[Text]:

return [

"OC\_BOYS",

"OC\_GIRLS",

"BC\_A\_BOYS",

"BC\_A\_GIRLS",

"BC\_B\_BOYS",

"BC\_B\_GIRLS",

"BC\_C\_BOYS",

"BC\_C\_GIRLS",

"BC\_D\_BOYS",

"BC\_D\_GIRLS",

"BC\_E\_BOYS",

"BC\_E\_GIRLS",

"SC\_BOYS",

"SC\_GIRLS",

"ST\_BOYS",

"ST\_GIRLS",

]

def validate\_caste(

self,

value: Text,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> Dict[Text, Any]:

if value.upper() in self.caste\_db():

return {"caste": value}

else:

dispatcher.utter\_message(template="utter\_wrong\_caste")

return {"caste": None}

@staticmethod

def validate\_rank(

value: Text,

dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any],

) -> Dict[Text, Any]:

if len(value) <= 6:

return {"rank": value}

else:

dispatcher.utter\_message(template="utter\_wrong\_rank")

return {"rank": None}

# accessing data from json file

class ActionFinalWorld(Action):

def name(self) -> Text:

return "action\_output"

def run(self, dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any]

) -> List[Dict[Text, Any]]:

new\_c = tracker.get\_slot("caste")

new\_r = tracker.get\_slot("rank")

new\_co = tracker.get\_slot("course")

f = open('csvjson.json', )

data = json.load(f)

for i in data['college\_prediction']:

if i[new\_c] == 'NA':

pass

else:

x = i[new\_c]

if (int(x) - 5000 < int(new\_r)) and (int(x) + 10000 > int(new\_r)) and (i["Branch\_Code"] == new\_co):

dispatcher.utter\_message("college name\t" + i['Inst\_Name'] + "\nfee\t" + str(i['Tuition\_Fee'])+ "\nLast Rank\t" + str(i[new\_c]))

else:

pass

f.close()

return []

class TopCollege(Action):

def name(self) -> Text:

return "action\_colleges"

def run(self, dispatcher: CollectingDispatcher,

tracker: Tracker,

domain: Dict[Text, Any]

) -> List[Dict[Text, Any]]:

dispatcher.utter\_message("1:JNTU HYDERABAD - JNTU ")

dispatcher.utter\_message("2:Osmania University - OU")

dispatcher.utter\_message("3:CBIT(Chaitanya Bharat Institute of technology) - OU(its autonomous college)")

dispatcher.utter\_message("4:Vasavi - OU""5:VNR Vignan Jyothi - JNTU")

dispatcher.utter\_message("6:SriNidhi College - JNTU")

dispatcher.utter\_message("7:Naraynamma college - JNTU")

dispatcher.utter\_message("8:Vardhaman college - Autonomus")

dispatcher.utter\_message("9:CVR College - JNTU")

dispatcher.utter\_message("10:Gokaraju Rangaraju - JNTU")

dispatcher.utter\_message("11:MVSR College - OU")

dispatcher.utter\_message("12:JNTU Kareemnagar - JNTU")

dispatcher.utter\_message("13:MGIT College - JNTU (Beside CBIT College)")

dispatcher.utter\_message("14:Muffakham Jah College of Engineering (MJ college best for minority - OU students)")

dispatcher.utter\_message("15:JNTU: Sultanpur - JNTU")

dispatcher.utter\_message("16:BVRIT College - JNTU")

dispatcher.utter\_message("17:Matrusri College - OU")

dispatcher.utter\_message("18:CVSR College - JNTU")

dispatcher.utter\_message("19:IARE College(best for aeronautical branch also other branches - JNTUare there)")

dispatcher.utter\_message("20:JNTU Manthani - JNTU")

return []