CSE 571: Web Services

Project III Report

Implementation of Coordination Protocol

Seema Sharanappa Kanaje

https://drive.google.com/file/d/1X2oE7Q8U7DiByAOXMLB8-1WWETwDKZFq/view?usp=sharing

Source code of the implemented webservices:

Here, I have implemented 10 REST API's and have "get" method for each of them and all of them are invoked by "Service" API according to co-ordination protocol. Each API gets its input from "Service" API. For instance, if "Service" API invokes "service1". "service1" API checks whether the username is between A-M or N-Z. If it is A-M then it returns Username, Invocation Date and Invocation Time in JSON format otherwise it returns in xml format. The same functionality is applied for all the REST API's.

```
class service1(Resource):
    def get(self,username):

        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"

            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service2(Resource):
    def get(self,username):

        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
```

```
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service3(Resource):
    def get(self,username):

        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
```

```
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service4(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service5(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)
```

```
JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service6(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
```

```
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service7(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service8(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
```

```
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"
class service9(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service10(Resource):
```

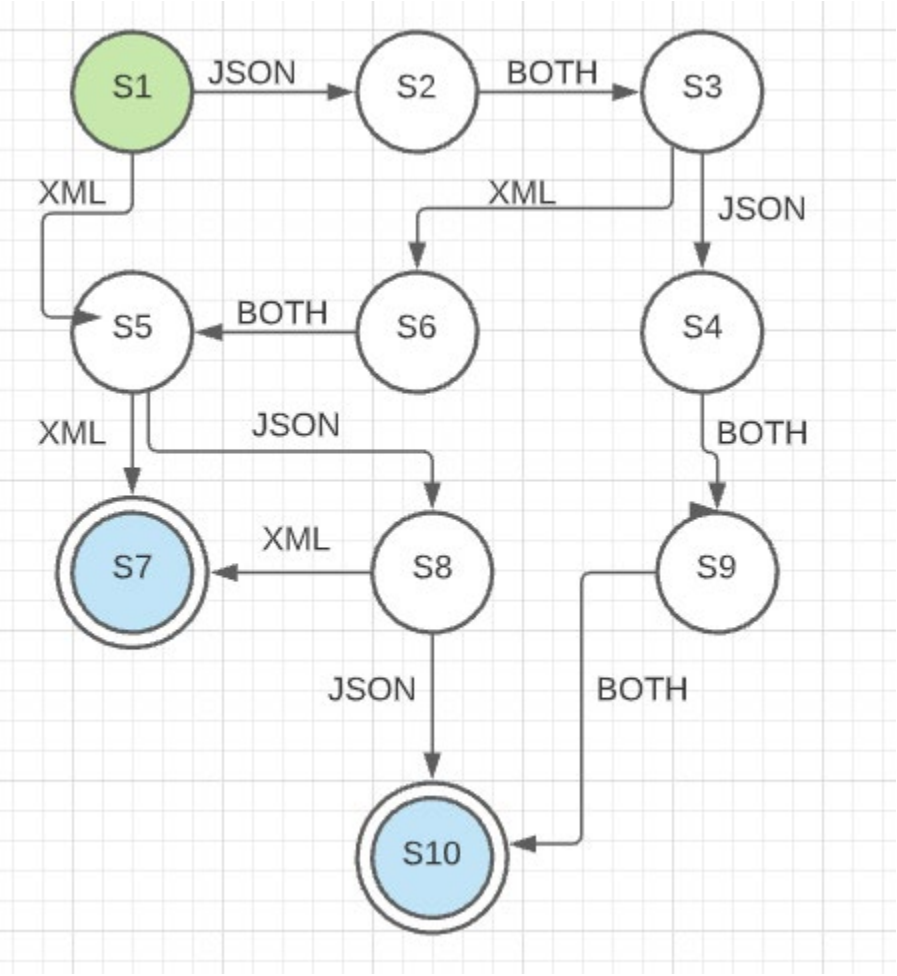
```
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()

JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+st
r(current_time.year)

JSON_Details['time']=str(current_time.hour)+"Hr"+str(current_time.minute)+"Mi
n"+str(current_time.second)+"Sec"
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" +
str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + "Hr" +
str(current_time.minute) + "Min" + str(current_time.second) + "Sec"

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time


            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"



api.add_resource(service, '/service')
api.add_resource(service1, '/service1/<username>')
api.add_resource(service2, '/service2/<username>')
api.add_resource(service3, '/service3/<username>')
api.add_resource(service4, '/service4/<username>')
api.add_resource(service5, '/service5/<username>')
api.add_resource(service6, '/service6/<username>')
api.add_resource(service7, '/service7/<username>')
api.add_resource(service8, '/service8/<username>')
api.add_resource(service9, '/service9/<username>')
api.add_resource(service10, '/service10/<username>')



app.run(port=5003)
```

A description of conversational handler:

Algorithm:

- Read Co-ordination Protocol in json format and store it in an array.
- Take inputs from front end your username and api name that needs to be invoked.
- Check if it's a start state by making a comparison with Co-ordination protocol.
    - If yes, store the next state based on its username input and then invoke the API.
    - Else, throw an error.
- Now user enters new API in the front end, and it validates if it matches with next state. If it is true, then it invokes the next API.
- We also check if its next state is final state it will display a message saying final state.
- Otherwise, displays a message saying "Error, input is not according to co-ordination protocol."

State Diagram for Co-ordination Protocol:



Co-ordination_Protocol JSON Format:


```
{
	"startstate":["service1"],
	"finalstate":["service7","service10"],
	"service1":{
		"json":"service2",
		"xml":"service5"
	},
	"service2":{
		"both":"service3"

	},
	"service3":{
		"json":"service4",
		"xml":"service6"
	},
	"service4":{
		"both":"service9"
	},
	"service5":{
		"json":"service8",
		"xml":"service7"
	},
	"service6":{
		"both":"service5"
	},
	"service8":{
		"json":"service10",
		"xml":"service7"
	},
```

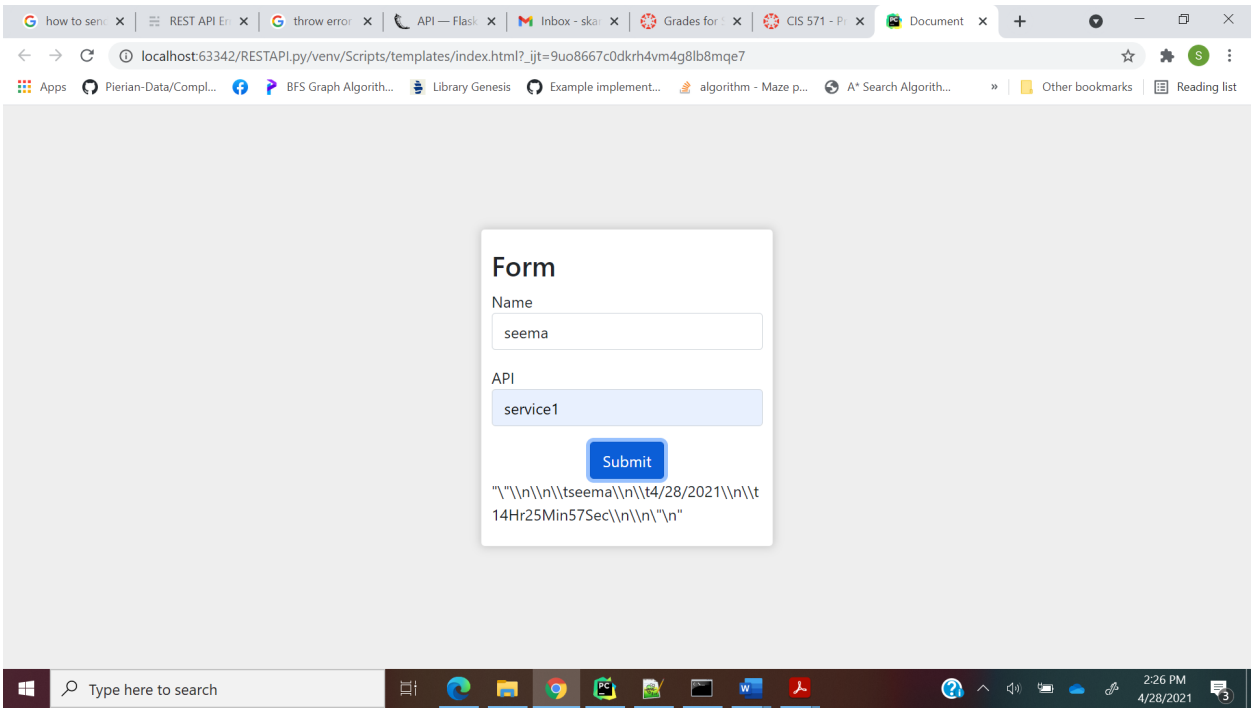
```
        "service9":{
                "both":"service10"
        }
}
```

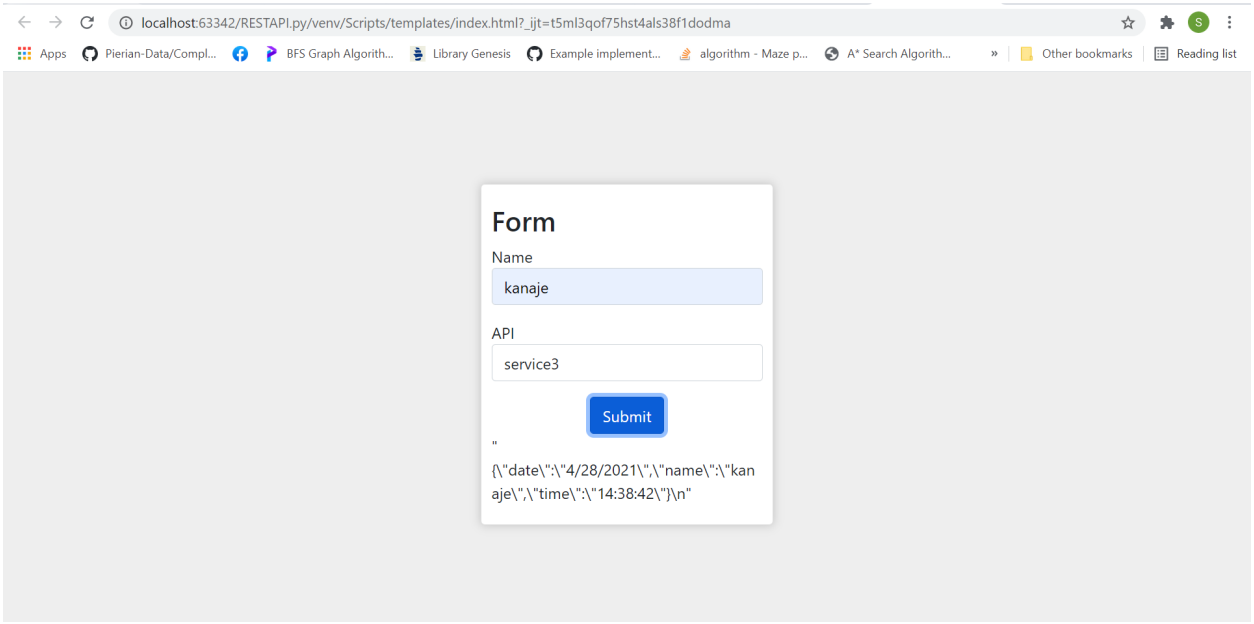
Source code for the Conversation Handler:

```
class service(Resource):
    def get(self):
        #print(data['service1']['json'])

        username = request.args.get('name')
        current_api=request.args.get('api')
        print(next_state)

        if(len(api_order_list)==0):
            print("After len")
            print(current_api)
            print(data['startstate'][0])

            if(current_api==data['startstate'][0]):

                api_order_list.append(current_api)
                if re.search(pattern, username[0]):
                    next_state.append(data[current_api]['json'])
                elif re.search(pattern1, username[0]):
                    next_state.append(data[current_api]['xml'])
                link = str('http://127.0.0.1:5003/' + current_api + '/' +
username)
                resp = requests.get(link)
                return resp.text

            else:
                return "Error, please start from initial state"
        elif (next_state[len(next_state)-1]==current_api and current_api not
in final_state):
            api_order_list.append(current_api)
            try:
                if re.search(pattern, username[0]):
                    next_state.append(data[current_api]['json'])
                elif re.search(pattern1, username[0]):
                    next_state.append(data[current_api]['xml'])
            except:
                next_state.append(data[current_api]['both'])
            link = str('http://127.0.0.1:5003/' + current_api + '/' +
username)
            resp = requests.get(link)
            print(next_state)
```

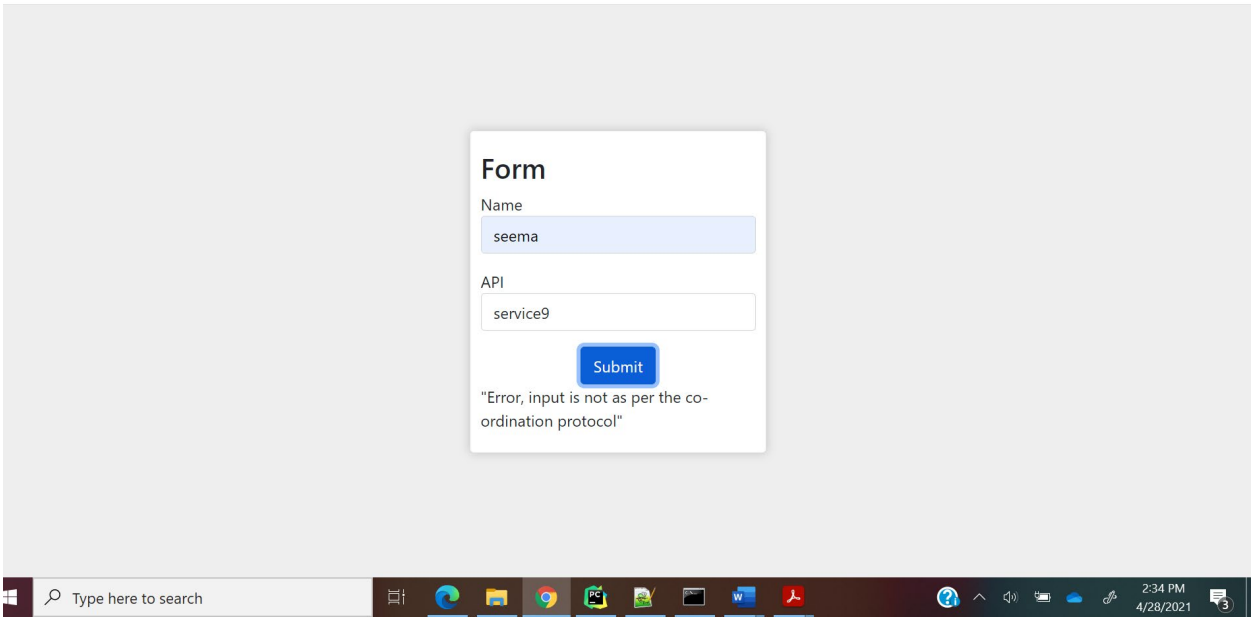
```
            return resp.text
        elif(current_api in final_state):

            return "Final State"
        else:

            return "Error, input is not as per the co-ordination protocol"
```

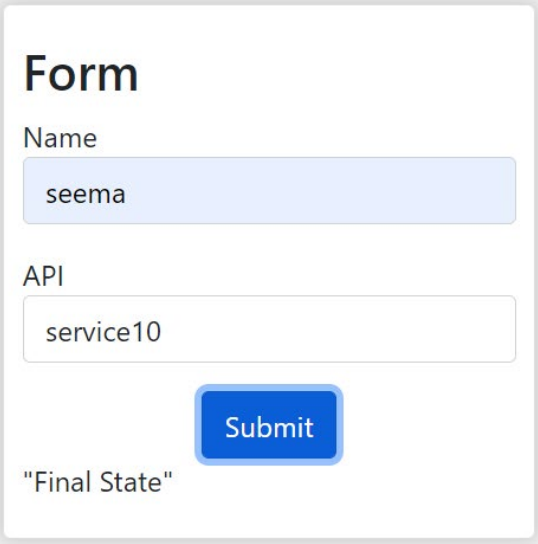
Screenshots of implementation:





Gives a below error when there API calls aren't made as per co-ordination protocol.


Form
Name
seema
API
service9
Submit
"Error, input is not as per the co-ordination protocol"
Type here to search
2:34 PM
4/28/2021


Form
Name
seema
API
service10
Submit
"Final State"

Libraries:

- flask
- re
- datetime
- json
- xml.etree.ElementTree
- xml.dom.minidom
- flask_cors
- requests
- flask_restplus
- werkzeug.utils

Implementation Files:

app.py

```
from flask import Flask, jsonify, request, render_template,abort
import re
import datetime
import json
import xml.etree.ElementTree as ET
import xml.dom.minidom
from flask_cors import CORS
import requests
from flask_restplus import Resource, Api
from werkzeug.utils import cached_property

app = Flask(__name__)
CORS(app)
api = Api(app)

pattern='[a-m]+'
pattern1='[n-z]+'
#username ='abc'
with open("coordinationprotocol.json", encoding="utf8") as file:
    data = json.load(file)
start_state = data['startstate'][0]
final_state = data['finalstate']
print(data)
api_order_list=[]
next_state=[]
print(api_order_list)
```

```
class service(Resource):
    def get(self):
        #print(data['service1']['json'])

        username = request.args.get('name')
        current_api=request.args.get('api')
        print(next_state)

        if(len(api_order_list)==0):
            print("After len")
            print(current_api)
            print(data['startstate'][0])

            if(current_api==data['startstate'][0]):

                api_order_list.append(current_api)
                if re.search(pattern, username[0]):
                    next_state.append(data[current_api]['json'])
                elif re.search(pattern1, username[0]):
                    next_state.append(data[current_api]['xml'])
                link = str('http://127.0.0.1:5003/' + current_api + '/' + username)
                resp = requests.get(link)
                return resp.text

            else:
                return "Error, please start from initial state"
        elif (next_state[len(next_state)-1]==current_api and current_api not in final_state):
            api_order_list.append(current_api)
            try:
                if re.search(pattern, username[0]):
                    next_state.append(data[current_api]['json'])
                elif re.search(pattern1, username[0]):
                    next_state.append(data[current_api]['xml'])
            except:
                next_state.append(data[current_api]['both'])
            link = str('http://127.0.0.1:5003/' + current_api + '/' + username)
            resp = requests.get(link)
            print(next_state)
            return resp.text
        elif(current_api in final_state):

            return "Final State"
        else:

            return "Error, input is not as per the co-ordination protocol"
```

```
#print(username[0])
JSON_Details={}
@app.route('/coordinationprotocol')
def signUp():
   return render_template('../templates/index.html')

class BadRequest(Exception):
   """Custom exception class to be thrown when local error occurs."""
   def __init__(self, message, status=400, payload=None):
      self.message = message
      self.status = status
      self.payload = payload


@app.errorhandler(BadRequest)

class service1(Resource):
   def get(self,username):

      print("Hey I am username",username)
      if re.search(pattern, username[0]): # in A-M return JSON
         JSON_Details['name']=username
         current_time=datetime.datetime.now()
         JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
         JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)

         return jsonify(JSON_Details)
      elif re.search(pattern1, username[0]):
         current_time = datetime.datetime.now()
         invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
         invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

         root = ET.Element("User_Details")
         ET.SubElement(root, 'username').text = username
         ET.SubElement(root, "invocation_date").text = invocation_date
         ET.SubElement(root, "invocation_date").text = invocation_time
```

```
            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service2(Resource):
    def get(self,username):

        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service3(Resource):
    def get(self,username):

        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
```

```
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service4(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time
```

```
            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service5(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service6(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
```

```
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service7(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
```

```
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"

class service8(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
            invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

            root = ET.Element("User_Details")
            ET.SubElement(root, 'username').text = username
            ET.SubElement(root, "invocation_date").text = invocation_date
            ET.SubElement(root, "invocation_date").text = invocation_time



            tree= ET.ElementTree(root)
            new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
            #print(new_xml)
            xmlnew = xml.dom.minidom.parseString(new_xml)
            xml_pretty_str = xmlnew.toprettyxml()
            return xml_pretty_str
        else:
            return "Error in your text, Text should be in between A-Z"
class service9(Resource):
    def get(self,username):
        print("Hey I am username",username)
        if re.search(pattern, username[0]): # in A-M return JSON
            JSON_Details['name']=username
            current_time=datetime.datetime.now()
            JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
            JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
            print(JSON_Details)
            return jsonify(JSON_Details)
        elif re.search(pattern1, username[0]):
            current_time = datetime.datetime.now()
            invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
```

```
        invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

        root = ET.Element("User_Details")
        ET.SubElement(root, 'username').text = username
        ET.SubElement(root, "invocation_date").text = invocation_date
        ET.SubElement(root, "invocation_date").text = invocation_time


        tree= ET.ElementTree(root)
        new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
        #print(new_xml)
        xmlnew = xml.dom.minidom.parseString(new_xml)
        xml_pretty_str = xmlnew.toprettyxml()
        return xml_pretty_str
    else:
        return "Error in your text, Text should be in between A-Z"

class service10(Resource):
  def get(self,username):
    print("Hey I am username",username)
    if re.search(pattern, username[0]): # in A-M return JSON
        JSON_Details['name']=username
        current_time=datetime.datetime.now()
        JSON_Details['date']=str(current_time.month)+"/"+str(current_time.day)+"/"+str(current_time.year)
        JSON_Details['time']=str(current_time.hour)+":"+str(current_time.minute)+":"+str(current_time.second)
        print(JSON_Details)
        return jsonify(JSON_Details)
    elif re.search(pattern1, username[0]):
        current_time = datetime.datetime.now()
        invocation_date = str(current_time.month) + "/" + str(current_time.day) + "/" + str(current_time.year)
        invocation_time = str(current_time.hour) + ":" + str(current_time.minute) + ":" + str(current_time.second)

        root = ET.Element("User_Details")
        ET.SubElement(root, 'username').text = username
        ET.SubElement(root, "invocation_date").text = invocation_date
        ET.SubElement(root, "invocation_date").text = invocation_time


        tree= ET.ElementTree(root)
        new_xml = ET.tostring(root, encoding='utf8').decode('utf8')
        #print(new_xml)
        xmlnew = xml.dom.minidom.parseString(new_xml)
        xml_pretty_str = xmlnew.toprettyxml()
        return xml_pretty_str
    else:
        return "Error in your text, Text should be in between A-Z"
```

```
api.add_resource(service, '/service')
api.add_resource(service1, '/service1/<username>')
api.add_resource(service2, '/service2/<username>')
api.add_resource(service3, '/service3/<username>')
api.add_resource(service4, '/service4/<username>')
api.add_resource(service5, '/service5/<username>')
api.add_resource(service6, '/service6/<username>')
api.add_resource(service7, '/service7/<username>')
api.add_resource(service8, '/service8/<username>')
api.add_resource(service9, '/service9/<username>')
api.add_resource(service10, '/service10/<username>')



app.run(port=5003)
```

index.html

```
< !DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
  <link rel="stylesheet" href="../static/css/style.css">
</head>
<body>
  <div class="card">
    <form class="d-flex flex-column" id="form">
      <h3>Form</h3>
      <div class="form-group">
        <label for="name">Name</label>
        <input type="text" class="form-control" id="name" placeholder="Enter name">
      </div>
      <div class="form-group mt-3 mb-3">
        <label for="api">API</label>
        <input type="text" class="form-control" id="api" placeholder="Enter API">
      </div>
      <button type="submit" class="btn btn-primary m-auto">Submit</button>
    </form>
```

```
    <span id="time"></span>

  </div>
  <script
  src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
  crossorigin="anonymous"></script>
  <script type="text/javascript" src="../static/js/helper.js"></script>
</body>
</html>
```

style.css

```
html, body {
  min-height: 100vh;
}

body {
  display: flex;
  align-items: center;
  justify-content: center;
  background: #EEE;
}

.card {
  display: flex;
  max-width: 300px;
  width: 100%;
```

```
 box-shadow: 0 0 10px rgba(0,0,0,0.2);
 background-color: #FFF;
 padding: 20px 10px;
}
```

helper.js

```
$(document).ready(function(e) {
 $('#form').submit(function(e) {
  e.preventDefault();
  var name = $('#name').val();
  var api = $('#api').val();
  if( !name || !api) {
   return;
  }

  // Need to change the url and the query params below, based on need
  $.ajax({
                                url: `http://127.0.0.1:5003/service?name=${name}&api=${api}`,
                                type: 'GET',
                                success: function(response, textStatus, request){
                                        console.log(response);

        var content_type=request.getResponseHeader('Content-Type')
        if(content_type.includes('json'))
        {
          $("#time").html(JSON.stringify(response))
        }
        else
        {
```

```
                $("#time").html(response)
        }

        console.log(request.getResponseHeader('Content-Type'));




                },


                error: function(error){
                        console.log(error);
                }
        });
 });
});
```