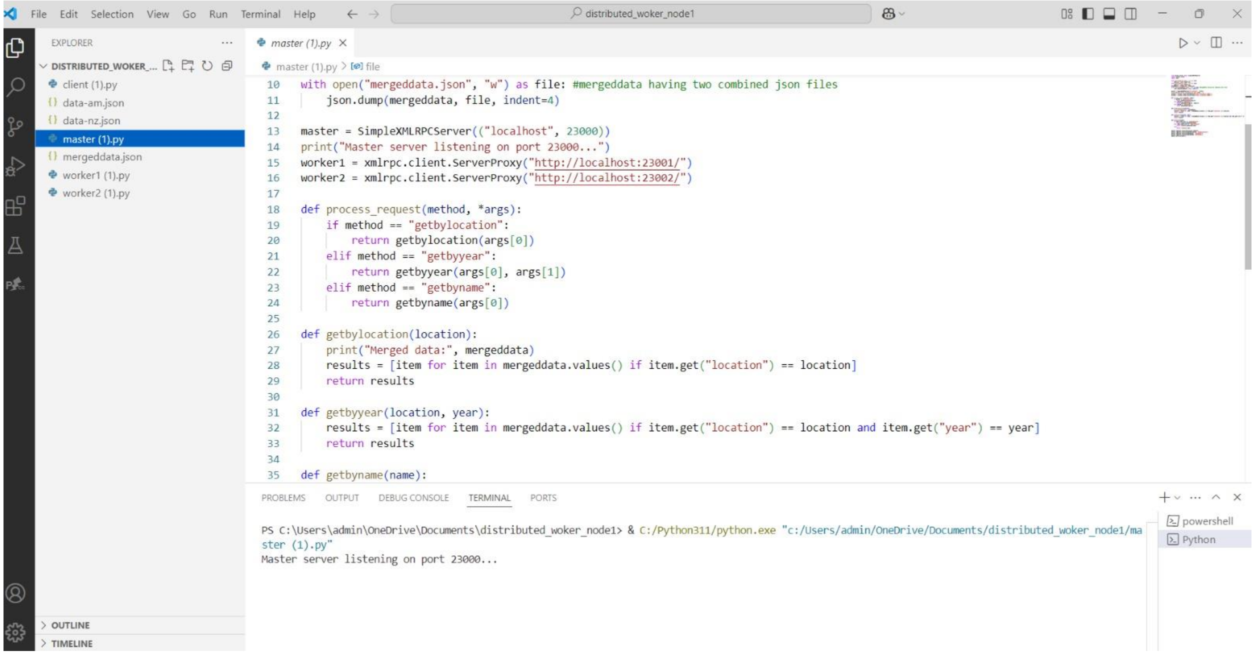
Programming Assignment –1

Samsritha Raavi 16351883

1)Master

This code creates a master XML-RPC server on localhost at port 23000. It loads and merges data from two JSON files, "data-am.json" and "data-nz.json," into a single dictionary. The server exposes three functions: getbylocation, getbyyear, and getbyname to query the merged data by location, year, or name. Name queries are forwarded to one of two worker servers (on ports 23001 and 23002) based on the first letter of the name. The master server registers these functions and continuously handles incoming requests.

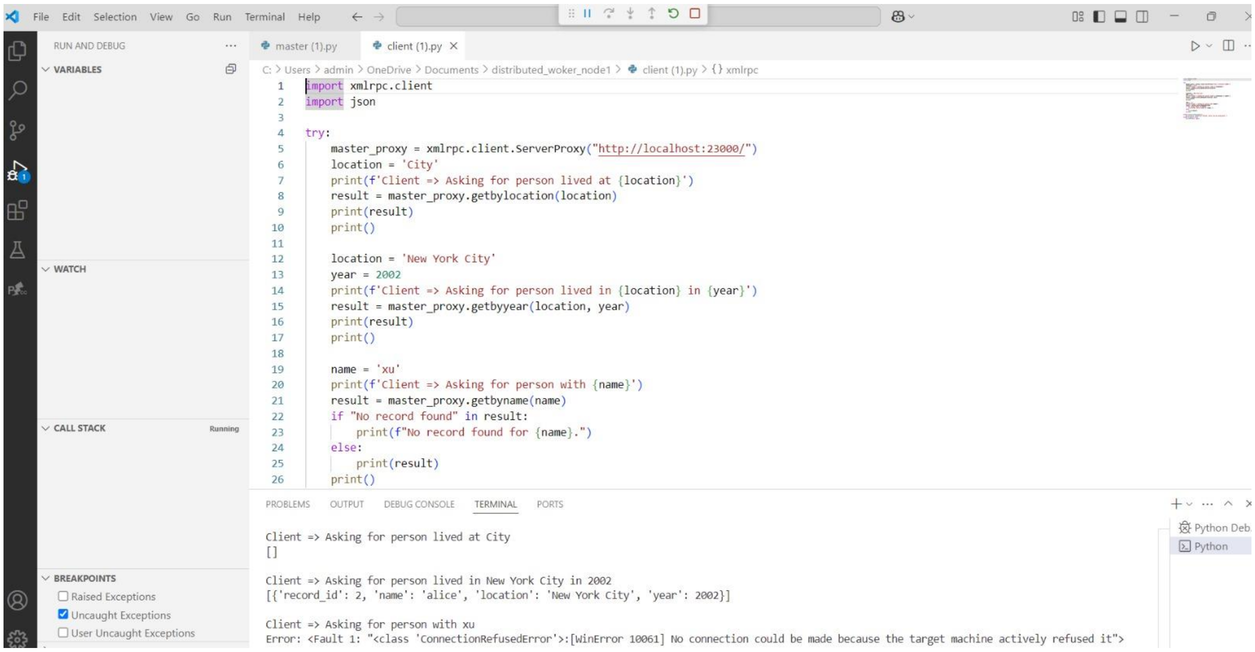


```python
with open("mergeddata.json", "w") as file: #mergeddata having two combined json files
    json.dump(mergeddata, file, indent=4)

master = SimpleXMLRPCServer(("localhost", 23000))
print("Master server listening on port 23000...")
worker1 = xmlrpc.client.ServerProxy("http://localhost:23001/")
worker2 = xmlrpc.client.ServerProxy("http://localhost:23002/")

def process_request(method, *args):
    if method == "getbylocation":
        return getbylocation(args[0])
    elif method == "getbyyear":
        return getbyyear(args[0], args[1])
    elif method == "getbyname":
        return getbyname(args[0])

def getbylocation(location):
    print("Merged data:", mergeddata)
    results = [item for item in mergeddata.values() if item.get("location") == location]
    return results

def getbyyear(location, year):
    results = [item for item in mergeddata.values() if item.get("location") == location and item.get("year") == year]
    return results

def getbyname(name):
```

```
PS C:\Users\admin\OneDrive\Documents\distributed_woker_node1> & C:/Python311/python.exe "c:/Users/admin/OneDrive/Documents/distributed_woker_node1/ma
ster (1).py"
Master server listening on port 23000...
```

2) Client

This code is a client that interacts with the master XML-RPC server running on localhost at port 23000. It queries the server for data based on specific criteria: location, year, and name. First, it requests information about a person living at a given location. Then, it asks for a person who lived in a specific city during a certain year. Lastly, it searches for a person by name. If no record is found, the client prints a message indicating so. The code includes error handling for connection issues and general exceptions to ensure smooth operation and provide informative error messages.



C: > Users > admin > OneDrive > Documents > distributed_woker_node1 > client (1).py > {} xmlrpc

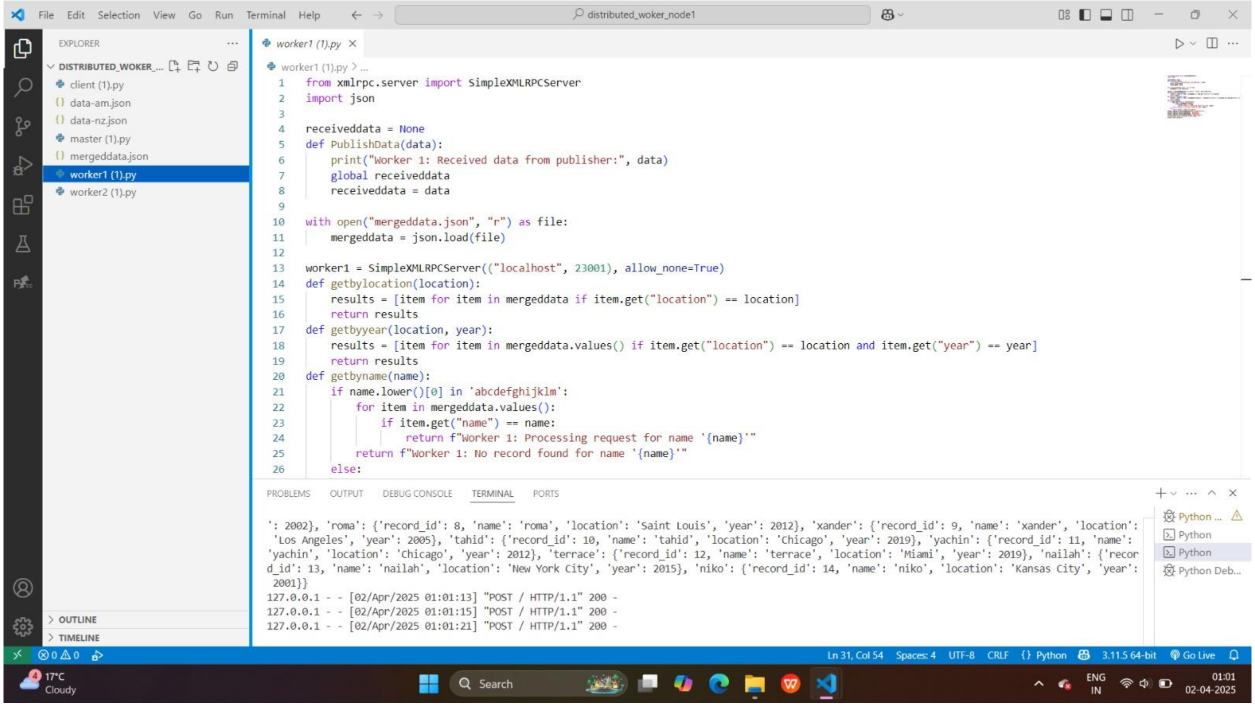
☐ Raised Exceptions
☑ Uncaught Exceptions
☐ User Uncaught Exceptions

```python
import xmlrpc.client
import json

try:
    master_proxy = xmlrpc.client.ServerProxy("http://localhost:23000/")
    location = 'City'
    print(f'Client => Asking for person lived at {location}')
    result = master_proxy.getbylocation(location)
    print(result)
    print()

    location = 'New York City'
    year = 2002
    print(f'Client => Asking for person lived in {location} in {year}')
    result = master_proxy.getbyyear(location, year)
    print(result)
    print()

    name = 'xu'
    print(f'Client => Asking for person with {name}')
    result = master_proxy.getbyname(name)
    if "No record found" in result:
        print(f"No record found for {name}.")
    else:
        print(result)
    print()
```

```
Client => Asking for person lived at City
[]

Client => Asking for person lived in New York City in 2002
[{'record_id': 2, 'name': 'alice', 'location': 'New York City', 'year': 2002}]

Client => Asking for person with xu
Error: <Fault 1: "<class 'ConnectionRefusedError'>:[WinError 10061] No connection could be made because the target machine actively refused it">
```

3) Worker1

This code sets up a worker XML-RPC server on localhost at port 23001. It defines four functions: getbylocation, getbyyear, getbyname, and PublishData. The server listens for requests to fetch data from a merged JSON file (mergeddata.json) based on location, year, or name. The PublishData function allows the worker to receive and store data from a publisher. The getbyname function processes requests for names starting with letters A-M; others are rejected. The server continuously runs and waits for incoming XML-RPC requests, processing them as needed.

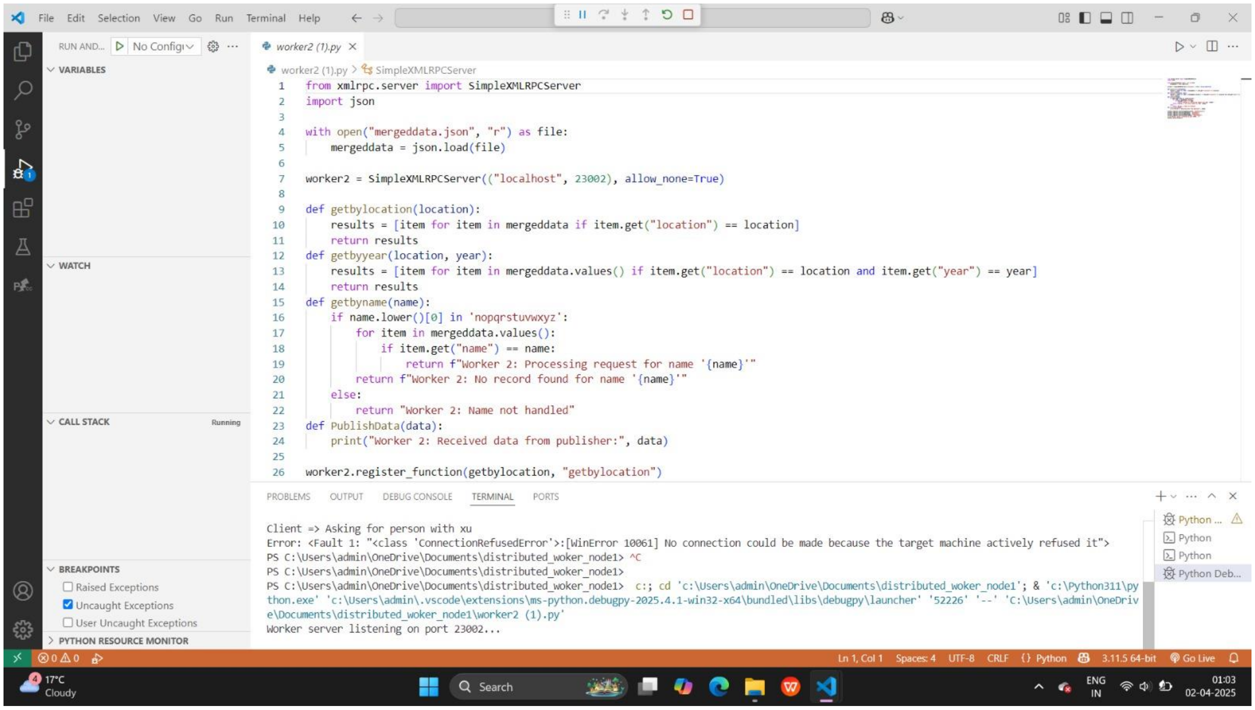


```python
from xmlrpc.server import SimpleXMLRPCServer
import json

receiveddata = None
def PublishData(data):
    print("Worker 1: Received data from publisher:", data)
    global receiveddata
    receiveddata = data

with open("mergeddata.json", "r") as file:
    mergeddata = json.load(file)

worker1 = SimpleXMLRPCServer(("localhost", 23001), allow_none=True)
def getbylocation(location):
    results = [item for item in mergeddata if item.get("location") == location]
    return results
def getbyyear(location, year):
    results = [item for item in mergeddata.values() if item.get("location") == location and item.get("year") == year]
    return results
def getbyname(name):
    if name.lower()[0] in 'abcdefghijklm':
        for item in mergeddata.values():
            if item.get("name") == name:
                return f"Worker 1: Processing request for name '{name}'"
        return f"Worker 1: No record found for name '{name}'"
    else:
```

```
': 2002}, 'roma': {'record_id': 8, 'name': 'roma', 'location': 'Saint Louis', 'year': 2012}, 'xander': {'record_id': 9, 'name': 'xander', 'location':
 'Los Angeles', 'year': 2005}, 'tahid': {'record_id': 10, 'name': 'tahid', 'location': 'Chicago', 'year': 2019}, 'yachin': {'record_id': 11, 'name':
'yachin', 'location': 'Chicago', 'year': 2012}, 'terrace': {'record_id': 12, 'name': 'terrace', 'location': 'Miami', 'year': 2019}, 'nailah': {'recor
d_id': 13, 'name': 'nailah', 'location': 'New York City', 'year': 2015}, 'niko': {'record_id': 14, 'name': 'niko', 'location': 'Kansas City', 'year':
 2001}}
127.0.0.1 - - [02/Apr/2025 01:01:13] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [02/Apr/2025 01:01:15] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [02/Apr/2025 01:01:21] "POST / HTTP/1.1" 200 -
```

4)Worker2

This code sets up a second worker XML-RPC server running on localhost at port 23002. It provides four functions: getbylocation, getbyyear, getbyname, and PublishData. These functions query data from the merged JSON file (mergeddata.json) based on location, year, or name. The getbyname function processes name queries starting with letters N-Z, while others are rejected. The PublishData function allows the server to receive data from a publisher and prints it out. The server listens for incoming XML-RPC requests and continuously serves requests indefinitely, handling queries as needed.



```python
from xmlrpc.server import SimpleXMLRPCServer
import json

with open("mergeddata.json", "r") as file:
    mergeddata = json.load(file)

worker2 = SimpleXMLRPCServer(("localhost", 23002), allow_none=True)

def getbylocation(location):
    results = [item for item in mergeddata if item.get("location") == location]
    return results
def getbyyear(location, year):
    results = [item for item in mergeddata.values() if item.get("location") == location and item.get("year") == year]
    return results
def getbyname(name):
    if name.lower()[0] in 'nopqrstuvwxyz':
        for item in mergeddata.values():
            if item.get("name") == name:
                return f"Worker 2: Processing request for name '{name}'"
        return f"Worker 2: No record found for name '{name}'"
    else:
        return "Worker 2: Name not handled"
def PublishData(data):
    print("Worker 2: Received data from publisher:", data)

worker2.register_function(getbylocation, "getbylocation")
```

```
Client => Asking for person with xu
Error: <Fault 1: "<class 'ConnectionRefusedError'>:[WinError 10061] No connection could be made because the target machine actively refused it">
PS C:\Users\admin\OneDrive\Documents\distributed_woker_node1> ^C
PS C:\Users\admin\OneDrive\Documents\distributed_woker_node1>
PS C:\Users\admin\OneDrive\Documents\distributed_woker_node1>  c:; cd 'c:\Users\admin\OneDrive\Documents\distributed_woker_node1'; & 'c:\Python311\py
thon.exe' 'c:\Users\admin\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '52226' '--' 'C:\Users\admin\OneDriv
e\Documents\distributed_woker_node1\worker2 (1).py'
Worker server listening on port 23002...
```