**Name -** Nalawade Prafulla Chandrashekhar.

**Class –** BE Artificial Intelligence and Data Science.

**Roll No. –** 42.

**Practical No. 01 -** Design a distributed application using RPC for remote computation where client submits an integer value to the server and server calculates factorial and returns the result to the client program.

**Factclient.py**

import xmlrpc.client

# Create an XML-RPC client

with xmlrpc.client.ServerProxy("http://localhost:8000/RPC2") as proxy:

    try:

        # Prompt the user to enter a number

        input\_value\_str = input("Enter the number: ")

        input\_value = int(input\_value\_str)

        # Call the server's calculate\_factorial method

        result = proxy.calculate\_factorial(input\_value)

        print(f"Factorial of {input\_value} is: {result}")

    except Exception as e:

        print(f"Error: {e}")

**Factserver.py**

from xmlrpc.server import SimpleXMLRPCServer

from xmlrpc.server import SimpleXMLRPCRequestHandler

class FactorialServer:

    def calculate\_factorial(self, n):

        if n < 0:

            raise ValueError("Input must be a non-negative integer.")

        result = 1

        for i in range(1, n + 1):

            result \*= i

        return result

# Restrict to a particular path

class RequestHandler(SimpleXMLRPCRequestHandler):

    rpc\_paths = ('/RPC2',)

# Create server

with SimpleXMLRPCServer(('localhost', 8000), requestHandler=RequestHandler) as server:

    server.register\_introspection\_functions()

    server.register\_instance(FactorialServer())

    print("FactorialServer is ready to accept requests.")

    server.serve\_forever()

**Output**

FactorialServer is ready to accept requests.

127.0.0.1 - - [17/Feb/2025 07:50:26] "POST /RPC2 HTTP/1.1" 200 –

Enter the number: 12

Factorial of 12 is: 479001600