NETWORK PROGRAMMING

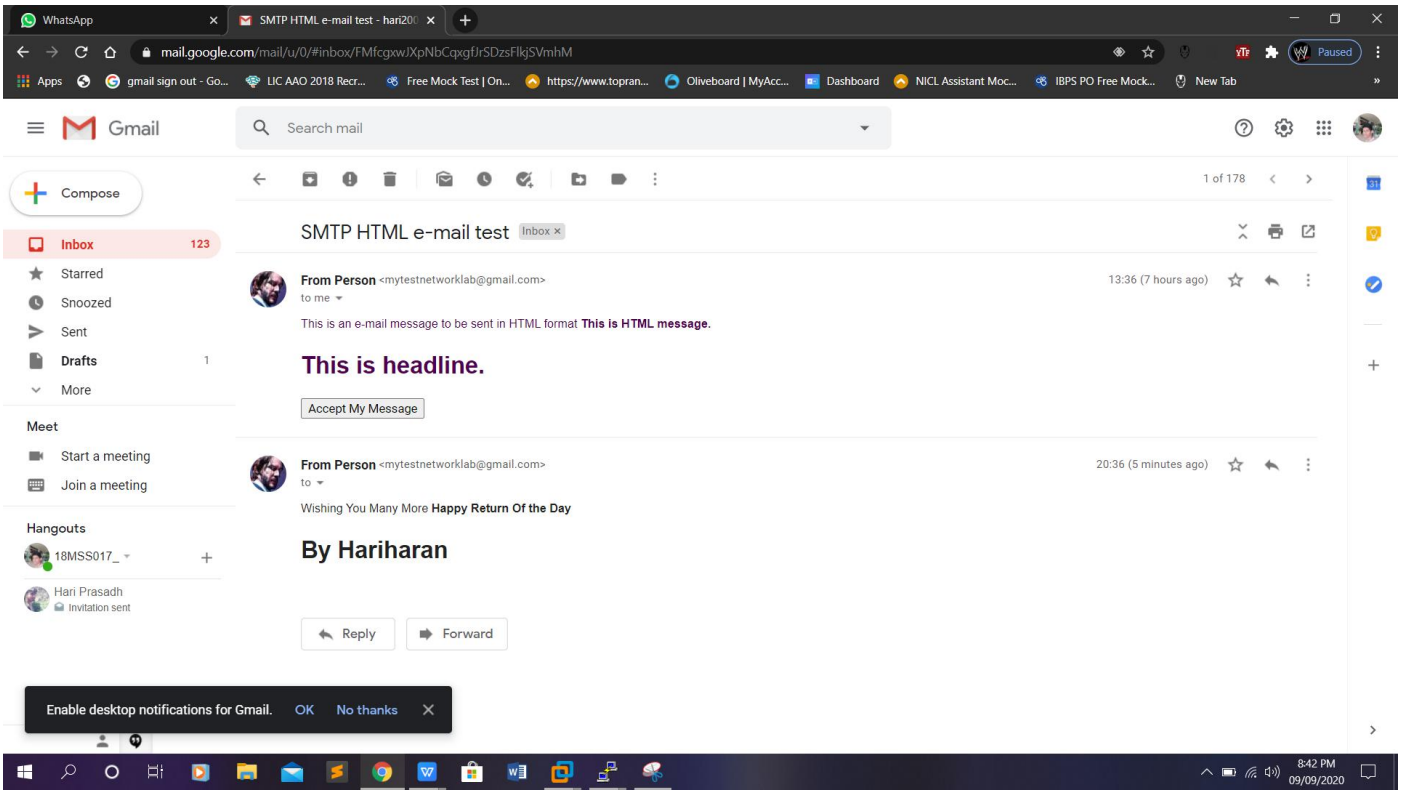
TASK-3

Name:S.Hariharan **Rollno:1831017**
Department:MSc Software Systems 5th Semester

1. Write a Java and Python program for sending a greeting mail to your friend using SMTP Protocol.

```
#!/usr/bin/python

import smtplib
sender = 'mytestnetworklab@gmail.com@gmail.com'
receivers = ['hari20010321@gmail.com','hari21032001@gmail.com']
message = """From: From Person <mytestnetworklab@gmail.com>
To: To Person
MIME-Version: 1.0
Content-type: text/html
Subject: SMTP HTML e-mail test

Wishing You Many More

<b>Happy Return Of the Day</b><br></br>
<h1>By Hariharan</h1>
"""
smtpObj = smtplib.SMTP('smtp.gmail.com',587)
smtpObj.ehlo()
smtpObj.starttls()
smtpObj.login("mytestnetworklab@gmail.com","test.1.2.3.4")
smtpObj.sendmail(sender, receivers, message)
print ("Successfully sent email")
smtpObj.quit()
```

Output:



Java Version:

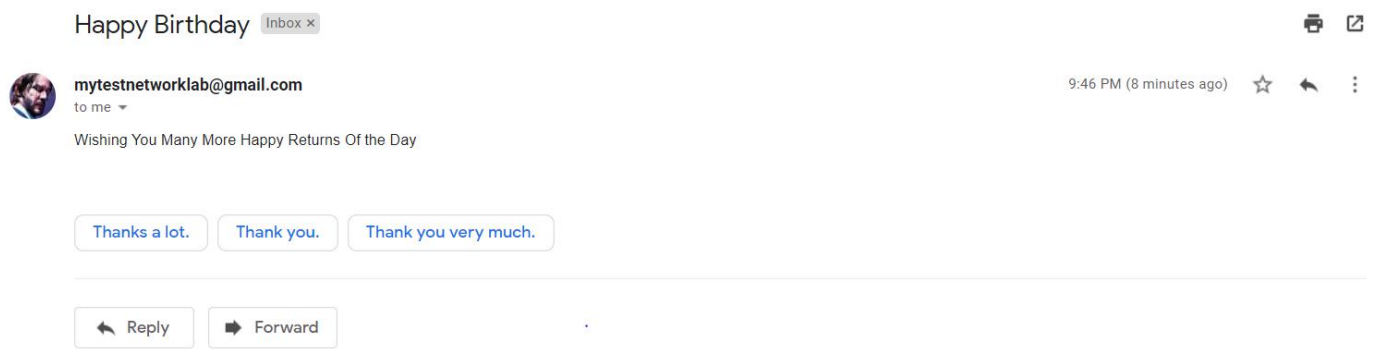
```
package sendmail;

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;
public class Sendmail {
        public static void main(String[] args) {
                // TODO code application logic here
                String to = "hari21032001@gmail.com";

// Add sender
  String from = "mytestnetworklab@gmail.com";
  final String username = "mytestnetworklab@gmail.com";//your Gmail username
  final String password = "test.1.2.3.4";//your Gmail password

String host = "smtp.gmail.com";
```

```
Properties props = new Properties();
  props.put("mail.smtp.auth", "true");
  props.put("mail.smtp.starttls.enable", "true");
  props.put("mail.smtp.host", host);
  props.put("mail.smtp.port", "587");

// Get the Session object
  Session session = Session.getInstance(props,
  new javax.mail.Authenticator() {
  protected PasswordAuthentication getPasswordAuthentication() {
  return new PasswordAuthentication(username, password);
  }
  });

try {
  // Create a default MimeMessage object
  Message message = new MimeMessage(session);

  message.setFrom(new InternetAddress(from));

  message.setRecipients(Message.RecipientType.TO,
  InternetAddress.parse(to));

  // Set Subject
  message.setSubject("Happy Birthday");

  // Put the content of your message
  message.setText("Wishing You Many More Happy Returns Of the Day");

// Send message
  Transport.send(message);

System.out.println("Sent message successfully....");

} catch (MessagingException e) {
  throw new RuntimeException(e);
  }
      }
```

}

Output:

```
run:
Sent message, successfully....
BUILD SUCCESSFUL (total time: 12 seconds)
```



2. Write a python program for sending Auto reply mail after receiving the greeting mail from ur friend using SMTP protocol.

Code:

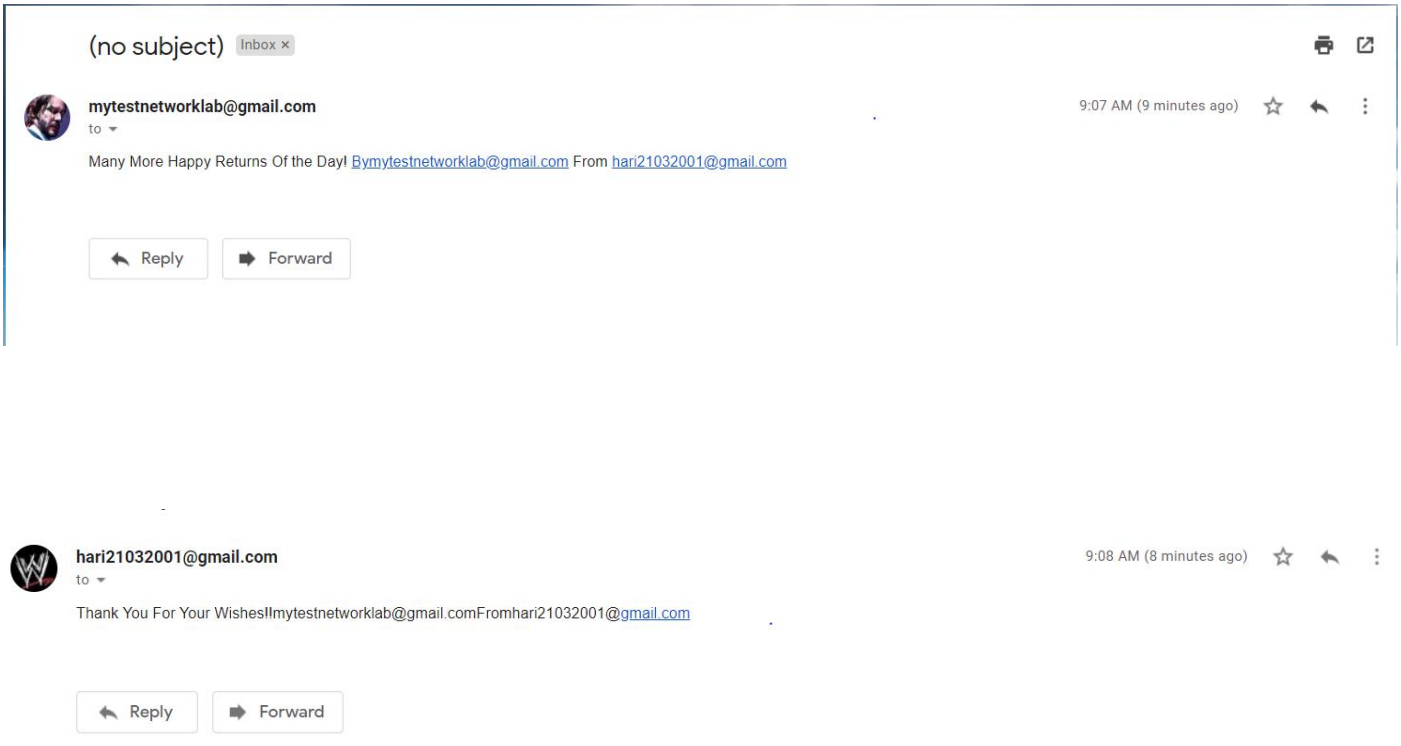
```
import smtplib

import getpass

# creates SMTP session

s1=smtplib.SMTP('smtp.gmail.com',587)

# start TLS for security

s1.starttls()

# Authentication

user1="mytestnetworklab@gmail.com"

user2="hari21032001@gmail.com"
```

```
print("Your email ID: "+user1)

password1=getpass.getpass(prompt='Enter your password:')

s1.login(user1,password1)

# message to be sent

message1="Many More Happy Returns Of the Day! By"+user1+" From "+user2

# sending the mail

print("Message sending in process....")

s1.sendmail(user1,user2,message1)

print("Message sent successfully from "+user1+" to "+user2)

# terminating the session

s1.quit()


if(user1=="mytestnetworklab@gmail.com"):

  s2=smtplib.SMTP('smtp.gmail.com',587)

# start TLS for security

  s2.starttls()
# Authentication

  password2="21.03.2001"

  s2.login(user2,password2)

# message to be sent
  message2="Thank You For Your Wishes!!"+user1+"From"+user2
```

```
#sending the mail

  print("Sending....")

  s2.sendmail(user2,user1,message2)

  print("Message sent to"+user1+" from "+user2)

# terminating the session

  s2.quit()
```

Output:

```
hariharan@hariharan:~/python$ python3 Automail.py
Your email ID: mytestnetworklab@gmail.com
Enter your password:
Message sending in process....
Message sent successfully from mytestnetworklab@gmail.com to hari21032001@gmail.com
Sending....
Message sent tomytestnetworklab@gmail.com from hari21032001@gmail.com
```



3.Write a Java and Python program for Two way socket communication Programming

Client.py:

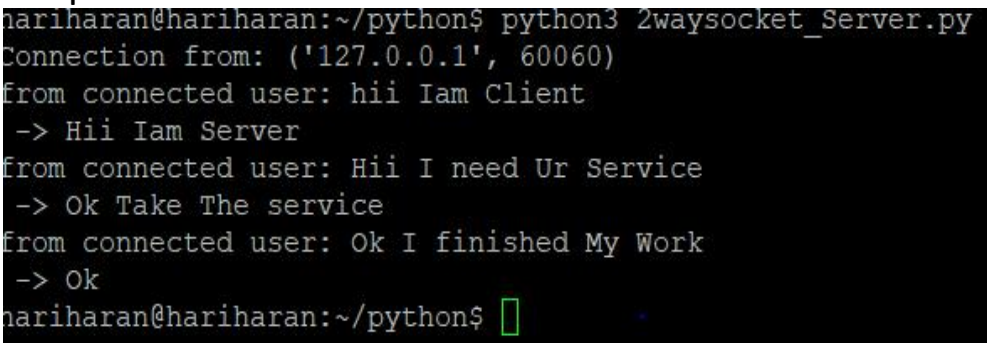
```
import socket
def client_program():
	host = socket.gethostname()  # as both code is running on same pc
	port = 5000  # socket server port number

	client_socket = socket.socket()  # instantiate
	client_socket.connect((host, port))  # connect to the server

	message = input(" -> ")  # take input

	while message.lower().strip() != 'bye':
		client_socket.send(message.encode())  # send message
		data = client_socket.recv(1024).decode()  # receive response

		print('Received from server: ' + data)  # show in terminal

		message = input(" -> ")  # again take input

	client_socket.close()  # close the connection


if __name__ == '__main__':
	client_program()
```

Server.Py:

```
import socket


def server_program():
	# get the hostname
	host = socket.gethostname()
	port = 5000  # initiate port no above 1024
```

```
    server_socket = socket.socket()  # get instance
    # look closely. The bind() function takes tuple as argument
    server_socket.bind((host, port))  # bind host address and port together

    # configure how many client the server can listen simultaneously
    server_socket.listen(2)
    conn, address = server_socket.accept()  # accept new connection
    print("Connection from: " + str(address))
    while True:
        # receive data stream. it won't accept data packet greater than 1024 bytes
        data = conn.recv(1024).decode()
        if not data:
            # if data is not received break
            break
        print("from connected user: " + str(data))
        data = input(' -> ')
        conn.send(data.encode())  # send data to the client

    conn.close()  # close the connection


if __name__ == '__main__':
    server_program()
```

Output:



```
hariharan@hariharan:~/python$ vi 2waysocket_Client.py
hariharan@hariharan:~/python$ python3 2waysocket_Client.py
 -> hii Iam Client
Received from server: Hii Iam Server
 -> Hii I need Ur Service
Received from server: Ok Take The service
 -> Ok I finished My Work
Received from server: Ok
 -> Bye
hariharan@hariharan:~/python$
```

Java Version:

Client.java:

```
import java.io.*;
import java.net.*;

class Client2Way {

	public static void main(String args[])
	throws Exception
	{
				Socket s = new Socket("localhost", 5555);

		DataOutputStream dos
			= new DataOutputStream(
				s.getOutputStream());

		BufferedReader br
			= new BufferedReader(
				new InputStreamReader(
					s.getInputStream()));

		BufferedReader kb
			= new BufferedReader(
				new InputStreamReader(System.in));
		String str, str1;

		while (!(str = kb.readLine()).equals("exit")) {

			dos.writeBytes(str + "\n");
```

```
            str1 = br.readLine();
            System.out.println("SERVER:"+str1);
            System.out.print("->");
        }

        dos.close();
        br.close();
        kb.close();
        s.close();
    }
}
```

SERVER.JAVA:

```
import java.io.*;
import java.net.*;

class Server2Way {

    public static void main(String args[])
        throws Exception
    {
        ServerSocket ss = new ServerSocket(5555);

        Socket s = ss.accept();
        System.out.println("Connection established");

        PrintStream ps
            = new PrintStream(s.getOutputStream());

        BufferedReader br
            = new BufferedReader(
                new InputStreamReader(
                    s.getInputStream()));

        BufferedReader kb
            = new BufferedReader(
                new InputStreamReader(System.in));

        while (true) {
```

```
                String str, str1;

                while ((str = br.readLine()) != null) {
                        System.out.println("CLIENT:"+str);
                        System.out.print("->");
                        str1 = kb.readLine();

                        ps.println(str1);
                }

                ps.close();
                br.close();
                kb.close();
                ss.close();
                s.close();

                System.exit(0);
        }
    }
}
```

```
hariharan@hariharan:~/java$ java Server2Way
Connection established
CLIENT:Hii Iam Client
->Welcome Client
CLIENT:I need Your Service
->Yeah Sure!!
CLIENT:Ok
->Ok done
CLIENT:Ok I'm Gonna Exit The Service
->Ok Done!! Bye!!
hariharan@hariharan:~/java$
```

```
hariharan@hariharan:~/java$ java Client2Way
Hii Iam Client
SERVER:Welcome Client
->I need Your Service
SERVER:Yeah Sure!!
->Ok
SERVER:Ok done
->Ok I'm Gonna Exit The Service
SERVER:Ok Done!! Bye!!
->exit
hariharan@hariharan:~/java$
```