

NAME: GVS SAI MADHAV

REG.NO: 19BCN7228

Q1: Implement all the methods of HTTP, use GUI to select any one method by the user

Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package httpapp;

/**
 *
 * @author saima
 */
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import org.jsoup.Jsoup;
import org.json.*;
import java.awt.Font;
import javax.swing.JButton;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.awt.event.ActionEvent;
import java.rmi.*;
import java.util.*;
public class HttpApp {

    private JFrame frame;
    private JTextField txtEnterUrl;
```

```

public JTextArea output;

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                HttpApp window = new HttpApp();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public HttpApp() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 10, 627, 753);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBackground(Color.CYAN);
    panel.setBounds(0, 0, 613, 716);
    frame.getContentPane().add(panel);
    panel.setLayout(null);

    JPanel panel_1 = new JPanel();
    panel_1.setBackground(new Color(255, 20, 147));
    panel_1.setBounds(0, 0, 623, 200);
    panel.add(panel_1);
    panel_1.setLayout(null);

    JLabel lblNewLabel = new JLabel("HTTP METHOD");
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 26));
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setTooltipText("");
    lblNewLabel.setBounds(199, 10, 253, 49);
    panel_1.add(lblNewLabel);

    JButton btnNewButton = new JButton("GET");

```

```

        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                try {
                    URL url =new URL(txtEnterUrl.getText());
                    HttpURLConnection
con=(HttpURLConnection)(url.openConnection());
                    System.out.println("WEBSITE :"+url.toString());
                    System.out.println("METHOD :"+con.getRequestMethod());
                    System.out.println("STATUS CODE :"+con.getResponseCode());
                    BufferedReader in=new BufferedReader(new
InputStreamReader(con.getInputStream()));

                    StringBuilder sb=new StringBuilder();
                    String line;
                    while((line=in.readLine())!=null) {
                        sb.append(line);
                    }
                    output.setText(Jsoup.parse(sb.toString()).html());

                } catch (Exception e1) {

                    output.setText(e1.toString());
                }

            }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 24));
        btnNewButton.setBounds(33, 69, 112, 37);
        panel_1.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("POST");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String text=output.getText();
                if(text.length()==0)
                {   output.setText("POST DATA CANNOT BE NULL");
                    return ;
                }
                if(text.split("=").length<=1) {
                    output.setText("Min of One Form Data is Needed");
                    return ;
                }
                try {

```

```

        String payload=output.getText();
        URL url=new URL(txtEnterUrl.getText());
        HttpURLConnection con=(HttpURLConnection)(url.openConnection());
        System.out.println("WEBSITE :"+url.toString());
        con.setRequestMethod("POST");
        System.out.println(con.getRequestMethod());
        con.setDoOutput(true);
        OutputStream os = con.getOutputStream();
        os.write(payload.getBytes());
        os.flush();
        os.close();
        if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
//success
            BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();

            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();

            // print result
            output.setText(response.toString());
            //JSONObject json =new JSONObject(response.toString());
            //output.setText(json.toString());
        } else {
            System.out.println("POST request not ALLOWED");
        }

    }catch(Exception e2) {
        output.setText(e2.toString());
    }

    }
});
btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 24));
btnNewButton_1.setBounds(171, 69, 112, 37);
panel_1.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("DELETE");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

String text=output.getText();
if(text.length()==0)
{
    output.setText("DELETE DATA CANNOT BE NULL");
    return ;
}
if(text.split("=").length<=1) {
    output.setText("Min of One Form Data is Needed");
    return ;
}
try {
String payload=output.getText();
URL url=new URL(txtEnterUrl.getText());
URLConnection con=(URLConnection)(url.openConnection());
System.out.println("WEBSITE :"+url.toString());
con.setRequestMethod("DELETE");
System.out.println(con.getRequestMethod());
con.setDoOutput(true);
OutputStream os = con.getOutputStream();
os.write(payload.getBytes());
os.flush();
os.close();
if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
    BufferedReader in = new BufferedReader(new InputStreamReader(
        con.getInputStream()));
    String inputLine;
    StringBuffer response = new StringBuffer();

    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
    in.close();
    output.setText(response.toString());

} else {
    System.out.println("DELETE request not ALLOWED");
}
}catch(Exception e2) {
    output.setText(e2.toString());
}

}

});
btnNewButton_2.setFont(new Font("Tahoma", Font.BOLD, 20));
btnNewButton_2.setBounds(304, 69, 112, 37);

```

```

panel_1.add(btnNewButton_2);

JButton btnNewButton_3 = new JButton("HEAD\r\n");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            URL url=new URL(txtEnterUrl.getText());
            System.out.println("WEBSITE :"+url.toString());
            HttpURLConnection con =
(HttpURLConnection)(url.openConnection());
            con.setRequestMethod("HEAD");
            System.out.println("METHOD :"+con.getRequestMethod());
            System.out.println("STATUS CODE :"+con.getResponseCode());
            output.setText("STATUS CODE
:"+con.getResponseCode()+"\n"+"CONTENT TYPE :"+con.getContentType());

            if(!(con.getResponseCode()>=200 && con.getResponseCode()<400
))
                output.setText("STATUS CODE
:"+con.getResponseCode()+"\nSITE IS DOWN !!!");

        } catch (Exception e1) {

            output.setText(e1.toString());

        }

    }
});
btnNewButton_3.setFont(new Font("Tahoma", Font.BOLD, 24));
btnNewButton_3.setBounds(443, 69, 112, 37);
panel_1.add(btnNewButton_3);

JButton btnNewButton_4 = new JButton("OPTIONS");
btnNewButton_4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            URL url =new URL(txtEnterUrl.getText());
            HttpURLConnection
con=(HttpURLConnection)(url.openConnection());
            con.setRequestMethod("OPTIONS");
            System.out.println("WEBSITE :"+url.toString());

```

```

        System.out.println("METHOD :"+con.getRequestMethod());
        System.out.println("STATUS CODE :"+con.getResponseCode());

        output.setText("ALLOWED METHODS
:"+con.getHeaderField("Allow"));

    } catch (Exception e1) {

        output.setText(e1.toString());
    }

}

});
btnNewButton_4.setFont(new Font("Tahoma", Font.BOLD, 16));
btnNewButton_4.setBounds(33, 129, 112, 37);
panel_1.add(btnNewButton_4);

JButton btnNewButton_5 = new JButton("PUT");
btnNewButton_5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        String text=output.getText();
        if(text.length()==0)
        {
            output.setText("PUT DATA CANNOT BE NULL");
            return ;
        }
        if(text.split("=").length<=1) {
            output.setText("Min of One Form Data is Needed");
            return ;
        }
        try {
            String payload=output.getText();
            URL url=new URL(txtEnterUrl.getText());
            HttpURLConnection con=(HttpURLConnection)(url.openConnection());
            System.out.println("WEBSITE :"+url.toString());
            con.setRequestMethod("PUT");
            System.out.println(con.getRequestMethod());
            con.setDoOutput(true);
            OutputStream os = con.getOutputStream();
            os.write(payload.getBytes());
            os.flush();
            os.close();

```

```

        if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
//success
            BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();

            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            output.setText(response.toString());

        } else {
            System.out.println("POST request not ALLOWED");
        }

    } catch (Exception e2) {
        output.setText(e2.toString());
    }

    }

});
btnNewButton_5.setFont(new Font("Tahoma", Font.BOLD, 24));
btnNewButton_5.setBounds(171, 129, 112, 37);
panel_1.add(btnNewButton_5);

JButton btnNewButton_6 = new JButton("TRACE");
btnNewButton_6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            URL url =new URL(txtEnterUrl.getText());
            HttpURLConnection
con=(HttpURLConnection)(url.openConnection());
            con.setRequestMethod("TRACE");
            System.out.println("WEBSITE :"+url.toString());
            System.out.println("METHOD :"+con.getRequestMethod());
            System.out.println("STATUS CODE :"+con.getResponseCode());

            StringBuilder sb=new StringBuilder();
            sb.append("\t\tHTTP Response Header\n");
            Map<String, List<String>> map = con.getHeaderFields();

```



```

        for (Map.Entry<String, List<String>> entry : map.entrySet())
        {
            sb.append(entry.getKey() + " : " +
entry.getValue()+"\n");
        }

        output.setText(sb.toString());

    } catch (Exception e1) {

        output.setText(e1.toString());
    }

    }
});
btnNewButton_6.setFont(new Font("Tahoma", Font.BOLD, 20));
btnNewButton_6.setBounds(304, 129, 112, 37);
panel_1.add(btnNewButton_6);

JPanel panel_2 = new JPanel();
panel_2.setBackground(new Color(0, 255, 255));
panel_2.setBounds(0, 199, 613, 517);
panel.add(panel_2);
panel_2.setLayout(null);

JPanel panel_3 = new JPanel();
panel_3.setBackground(new Color(244, 164, 96));
panel_3.setBounds(0, 0, 613, 66);
panel_2.add(panel_3);
panel_3.setLayout(null);

JLabel lblNewLabel_1 = new JLabel("URL");
lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 26));
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setBounds(36, 10, 116, 47);
panel_3.add(lblNewLabel_1);

txtEnterUrl = new JTextField();
txtEnterUrl.setFont(new Font("Tahoma", Font.BOLD, 16));
txtEnterUrl.setToolTipText("");
txtEnterUrl.setBounds(180, 10, 395, 39);
panel_3.add(txtEnterUrl);
txtEnterUrl.setColumns(10);

```

```
JPanel panel_4 = new JPanel();
panel_4.setBackground(Color.CYAN);
panel_4.setBounds(0, 62, 613, 455);
panel_2.add(panel_4);
panel_4.setLayout(null);

JLabel lblNewLabel_2 = new JLabel("RESPONSE");
lblNewLabel_2.setBounds(186, 10, 230, 39);
lblNewLabel_2.setFont(new Font("Tahoma", Font.BOLD, 24));
lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
panel_4.add(lblNewLabel_2);

output=new JTextArea();
output.setBounds(32,59, 541, 385);
JScrollPane scroll=new JScrollPane(output);
scroll.setBounds(32,59, 541, 385);
panel_4.add(scroll);

}
}
```

Output:

HTTP METHOD

GET

POST

DELETE

HEAD

OPTIONS

PUT

TRACE

URL

RESPONSE

