**WEB SERVER PROJECT**

**Professor**

Zhiyong Shan

**Class**

Saturday, 11:30 AM – 2:10 PM

**Group Members**

Subbaramireddy, Malkireddy (700639953)

Vijay Kumar, Thota (700647994)

Venkataramana Yashwant Kumar, Palisetty (700641099)

Pavana Krishna, Chunduru (700647307)

Pavan Kumar, Chaparala (700650622)

**WebServer.java Code:**



**package** com.ucmo.edu;

**import** java.io.BufferedReader;

**import** java.io.DataOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.InputStream;

**import** java.io.InputStreamReader;

**import** java.io.OutputStream;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**import** java.util.StringTokenizer;

**public** **class** WebServer **extends** Thread{

Socket socket=**null**;

DataOutputStream dataOutputStream=**null**;

**private** **static** ServerSocket *serverSocket*;

**public** WebServer(Socket acceptSocket) {

socket=acceptSocket;

}

**public** **static** **void** main(String args[]) **throws** Exception{

//Set port number

**int** port=9996;

*serverSocket* = **new** ServerSocket(port);

System.***out***.println ("TCPServer Waiting for client on port "+port);

//Process HTTP response in infinite loop

**do**

{

//Listen for TCP connection request

Socket acceptSocket=*serverSocket*.accept();

//Call the class constructor

WebServer webServer=**new** WebServer(acceptSocket);

//Start the thread

webServer.start();

}**while**(**true**);

}

//Within run(), we explicitly catch and handle exceptions with a try/catch block.

**public** **void** run(){

**try**{

processRequestFile();

}

**catch**(Exception e)

{

System.***out***.println(e);

}

}

**public** **void** processRequestFile() **throws** Exception{

System.***out***.println( "The Client "+ socket.getInetAddress() + ":" + socket.getPort() + " is connected");

// Get a reference to the socket's input and output streams.

InputStream inputStream=socket.getInputStream();

OutputStream outputStream=socket.getOutputStream();

dataOutputStream=**new** DataOutputStream(outputStream);

//Set up input stream filters.

InputStreamReader inputStreamReader=**new** InputStreamReader(inputStream);

//reads the input data

BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);

// Get the request line of the HTTP request message.

String requestString = bufferedReader.readLine();

// HERE WE NEED TO DEAL WITH THE REQUEST

// Extract the filename from the request line.

StringTokenizer tokenizer = **new** StringTokenizer(requestString);

// skip over the method, which should be "GET"

tokenizer.nextToken();

String httpQueryString = tokenizer.nextToken();

StringBuffer stringBuffer = **new** StringBuffer();

stringBuffer.append("<b> This is the HTTP Server Home Page.... </b><BR>The HTTP Client request is ....<BR>");

System.***out***.println("The HTTP request string is ....");

**while** (bufferedReader.ready())

{

// Read the HTTP complete HTTP Query

stringBuffer.append(requestString + "<BR>");

System.***out***.println(requestString);

requestString = bufferedReader.readLine();

}

**if** (httpQueryString.equals("/")) {

// The default home page

processResponseFile(200, stringBuffer.toString(), **false**);

} **else** {

//This is interpreted as a file name

String fileName = httpQueryString.replaceFirst("/", "");

//fileName = URLDecoder.decode(fileName);

**if** (**new** File(fileName).isFile()){

processResponseFile(200,fileName, **true**);

}

**else** {

System.***out***.println("filename isssss");

processResponseFile(404, "HTTP/1.0 404 Not Found", **false**);

}

}

}

**public** **void** processResponseFile (**int** statusCode, String responseString, **boolean** isFile) **throws** Exception {

//Construct the response message.

String statusLine = **null**;

String contentLengthLine = **null**;

String fileName = **null**;

String contentTypeLine = "Content-Type: text/html" + "\r\n"; //content info

FileInputStream fin = **null**;

**if** (statusCode == 200)

statusLine = "HTTP/1.1 200 OK" + "\r\n"; //common success message

**else**

statusLine = "HTTP/1.1 404 Not Found" + "\r\n"; //common error message

**if** (isFile) {

fileName = responseString;

fin = **new** FileInputStream(fileName);

contentLengthLine = "Content-Length: " + Integer.*toString*(fin.available()) + "\r\n";

**if** (!fileName.endsWith(".htm") && !fileName.endsWith(".html"))

contentTypeLine = "Content-Type: \r\n"; //content info

}

**else** {

responseString = "<html>" + "<title>HTTP Server in java</title>" + "<body>" + responseString + "</body>" + "</html>";

contentLengthLine = "Content-Length: " + responseString.length() + "\r\n";

}

dataOutputStream.writeBytes(statusLine);//Send the status line.

dataOutputStream.writeBytes("Server: Java HTTPServer");

dataOutputStream.writeBytes(contentTypeLine); //Send the content type line.

dataOutputStream.writeBytes(contentLengthLine); //Send the content length line.

dataOutputStream.writeBytes("Connection: close\r\n");

dataOutputStream.writeBytes("\r\n");

**if** (isFile) sendFile(fin, dataOutputStream);

**else** dataOutputStream.writeBytes(responseString);

dataOutputStream.close();

}

//set up input output streams

**public** **void** sendFile (FileInputStream fin, DataOutputStream out) **throws** Exception {

// Construct a 1K buffer to hold bytes on their way to the socket.

**byte**[] buffer = **new** **byte**[1024] ;

**int** bytesRead;

// Copy requested file into the socket's output stream.

// read() returns minus one, indicating that the end of the file

**while** ((bytesRead = fin.read(buffer)) != -1 ) {

out.write(buffer, 0, bytesRead);

}

fin.close();

}

}

**Index.html code:**



<!DOCTYPE html>

<html>

<head>

<title>WebServer Test1</title>

</head>

<body>

<h1>INDEX.HTML</h1>

<p>File found</p>

</body>

</html>

**Sample.html code:**



<!DOCTYPE html>

<html>

<head>

<title>WebServer Test2</title>

</head>

<body>

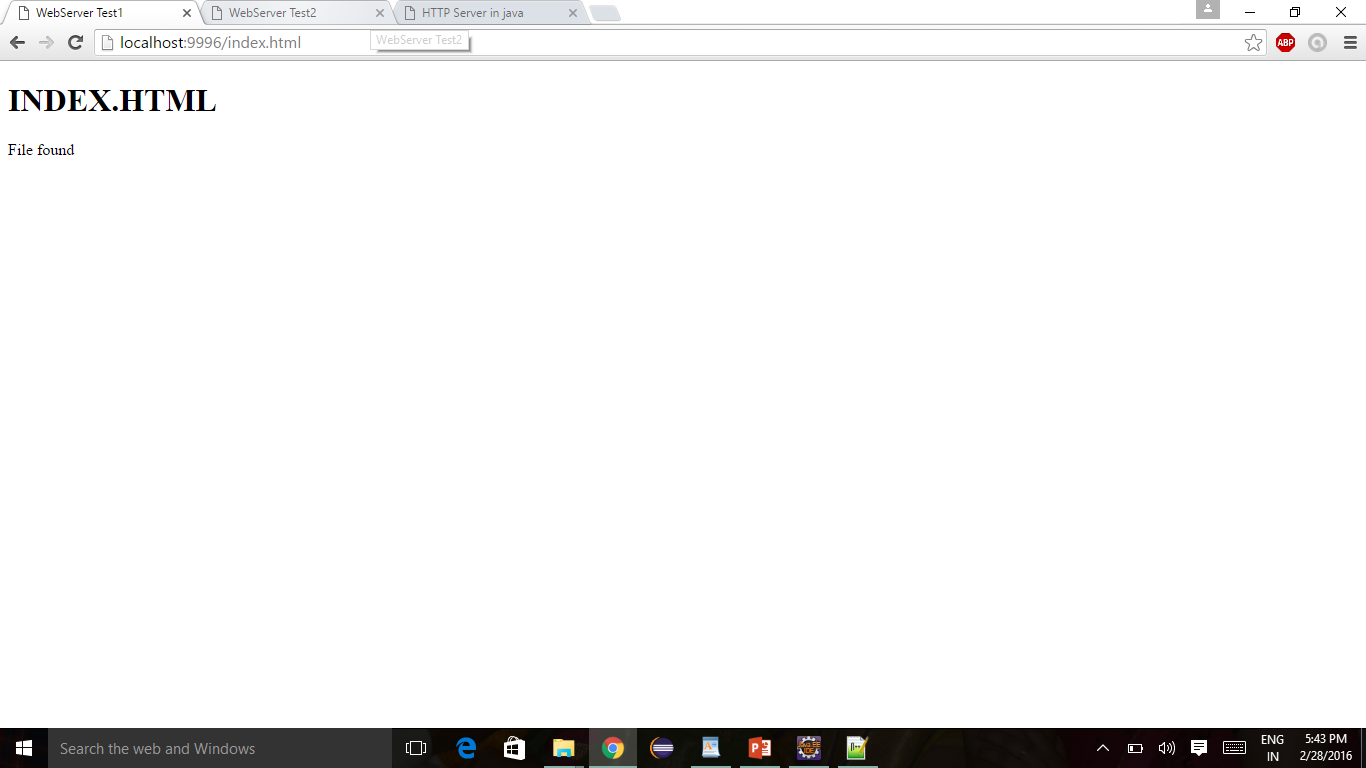
<h1>SAMPLE.HTML</h1>

<p>File found</p>

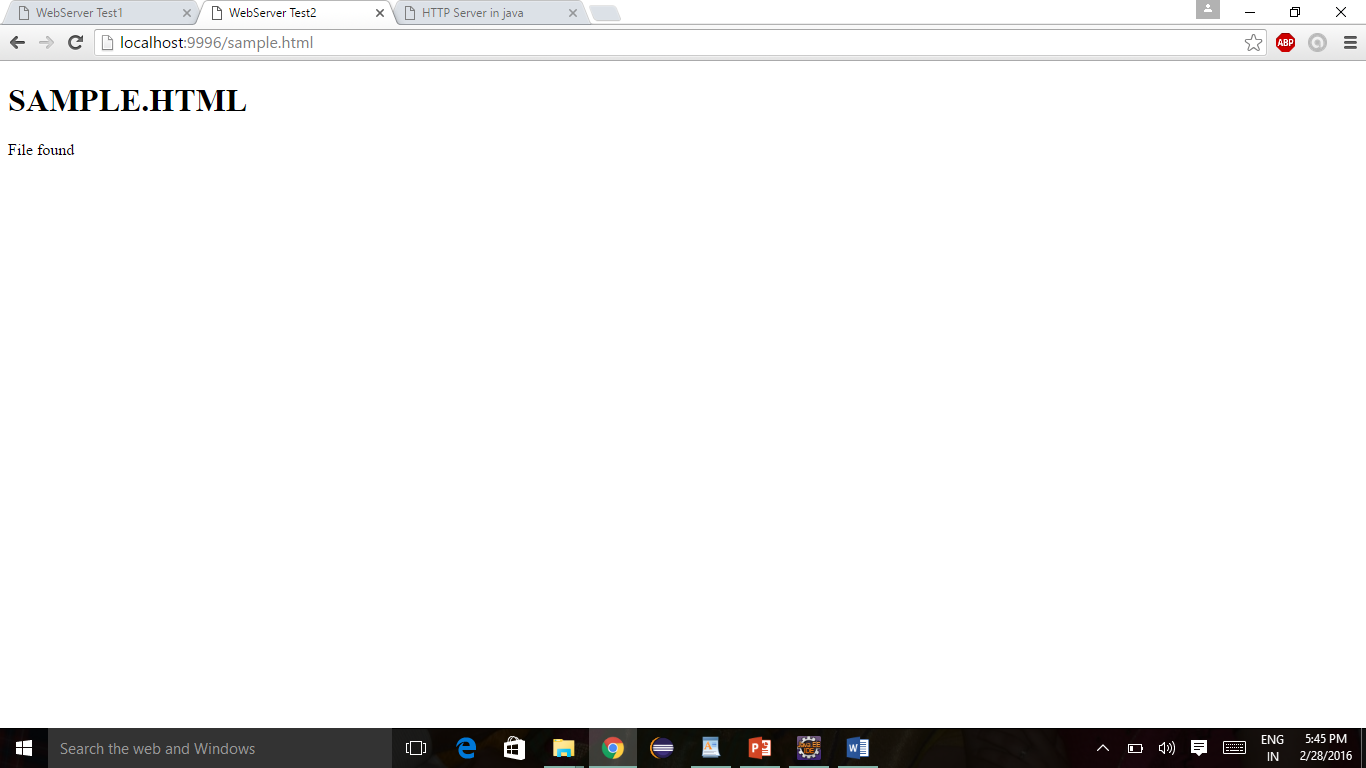
</body>

</html>

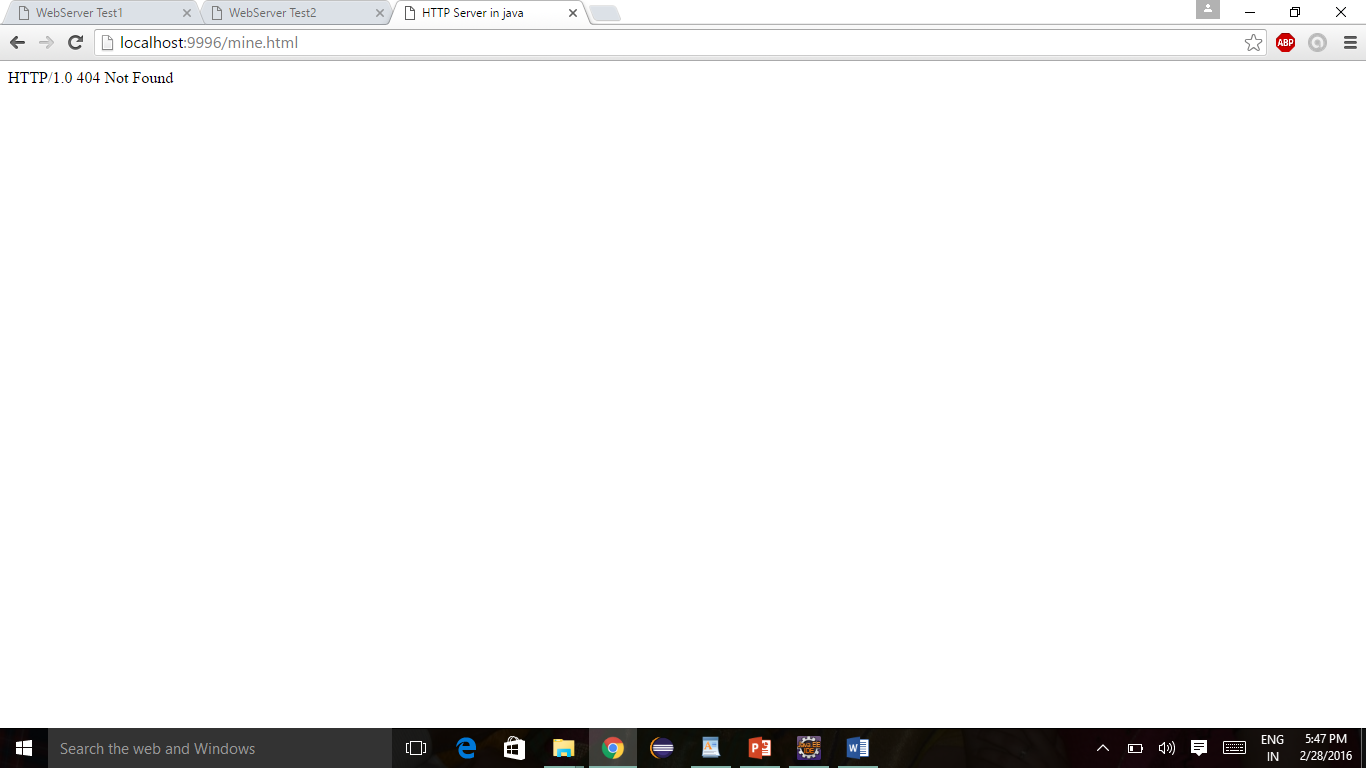
**Output after requesting “index.html” page:**



**Output after requesting “sample.html” page:**



**Output after requesting page that is not available in server:**



**Project ZIP file:**

****