

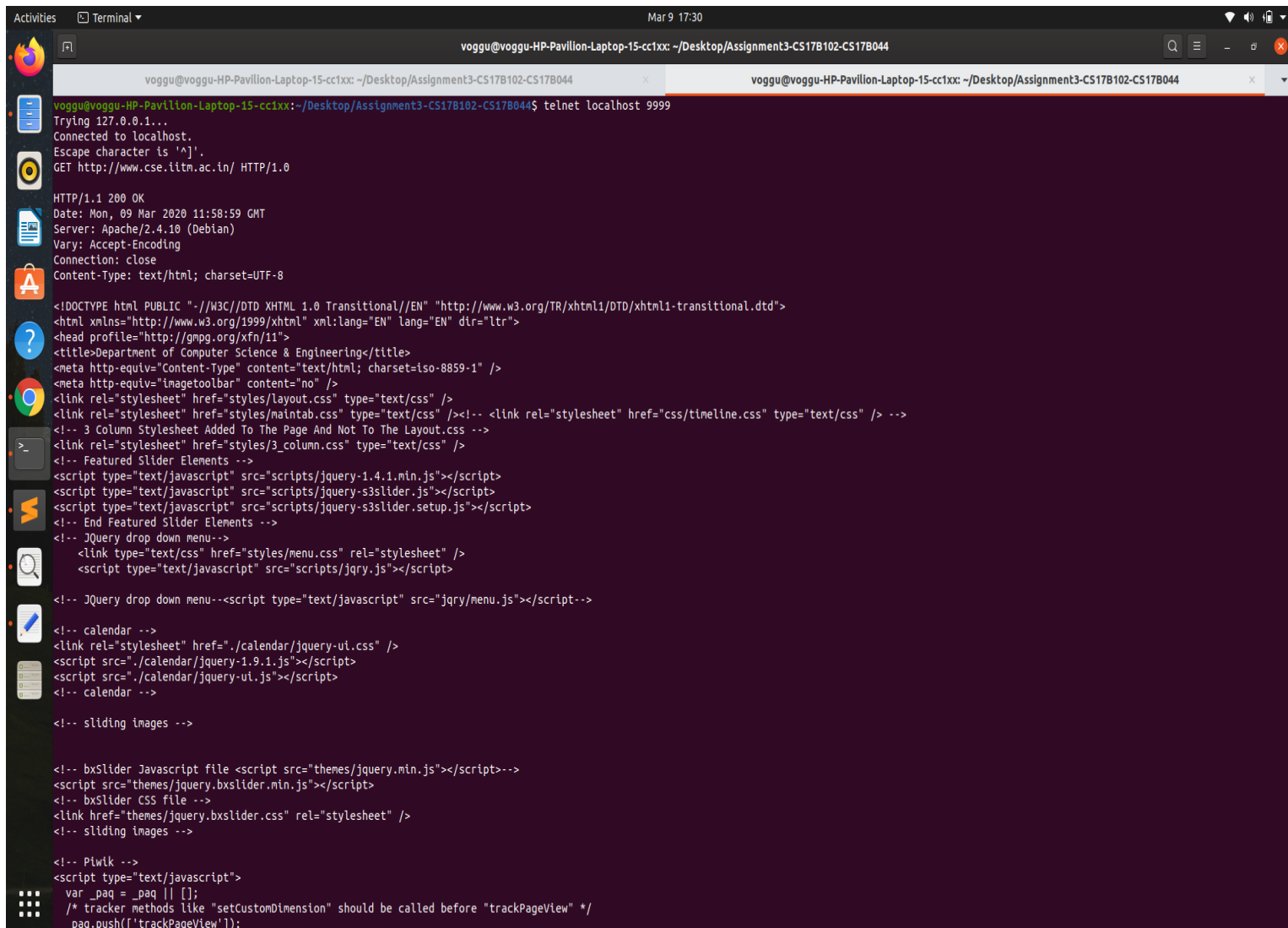
# ASSIGNMENT-3

## Implementing HTTP Web Proxy

CS17B044 V.Vamshi

CS17B102 B.Tarun

### Experiment 1: One Client (Single Terminal)



```
voggu@voggu-HP-Pavillon-Laptop-15-cc1xx: ~/Desktop/Assignment3-CS17B102-CS17B044
voggu@voggu-HP-Pavillon-Laptop-15-cc1xx:~/Desktop/Assignment3-CS17B102-CS17B044$ telnet localhost 9999
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
GET http://www.cse.iitn.ac.in/ HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 09 Mar 2020 11:58:59 GMT
Server: Apache/2.4.18 (Debian)
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="EN" lang="EN" dir="ltr">
<head profile="http://gmpg.org/xfn/11">
<title>Department of Computer Science & Engineering</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="imagetoolbar" content="no" />
<link rel="stylesheet" href="styles/layout.css" type="text/css" />
<link rel="stylesheet" href="styles/maintab.css" type="text/css" /><!-- <link rel="stylesheet" href="css/timeline.css" type="text/css" /> -->
<!-- 3 Column Stylesheet Added To The Page And Not To The Layout.css -->
<link rel="stylesheet" href="styles/3_column.css" type="text/css" />
<!-- Featured Slider Elements -->
<script type="text/javascript" src="scripts/jquery-1.4.1.min.js"></script>
<script type="text/javascript" src="scripts/jquery-s3slider.js"></script>
<script type="text/javascript" src="scripts/jquery-s3slider.setup.js"></script>
<!-- End Featured Slider Elements -->
<!-- JQuery drop down menu-->
<link type="text/css" href="styles/menu.css" rel="stylesheet" />
<script type="text/javascript" src="scripts/jqry.js"></script>

<!-- JQuery drop down menu--<script type="text/javascript" src="jqry/menu.js"></script-->

<!-- calendar -->
<link rel="stylesheet" href="./calendar/jquery-ui.css" />
<script src="./calendar/jquery-1.9.1.js"></script>
<script src="./calendar/jquery-ui.js"></script>
<!-- calendar -->

<!-- sliding images -->

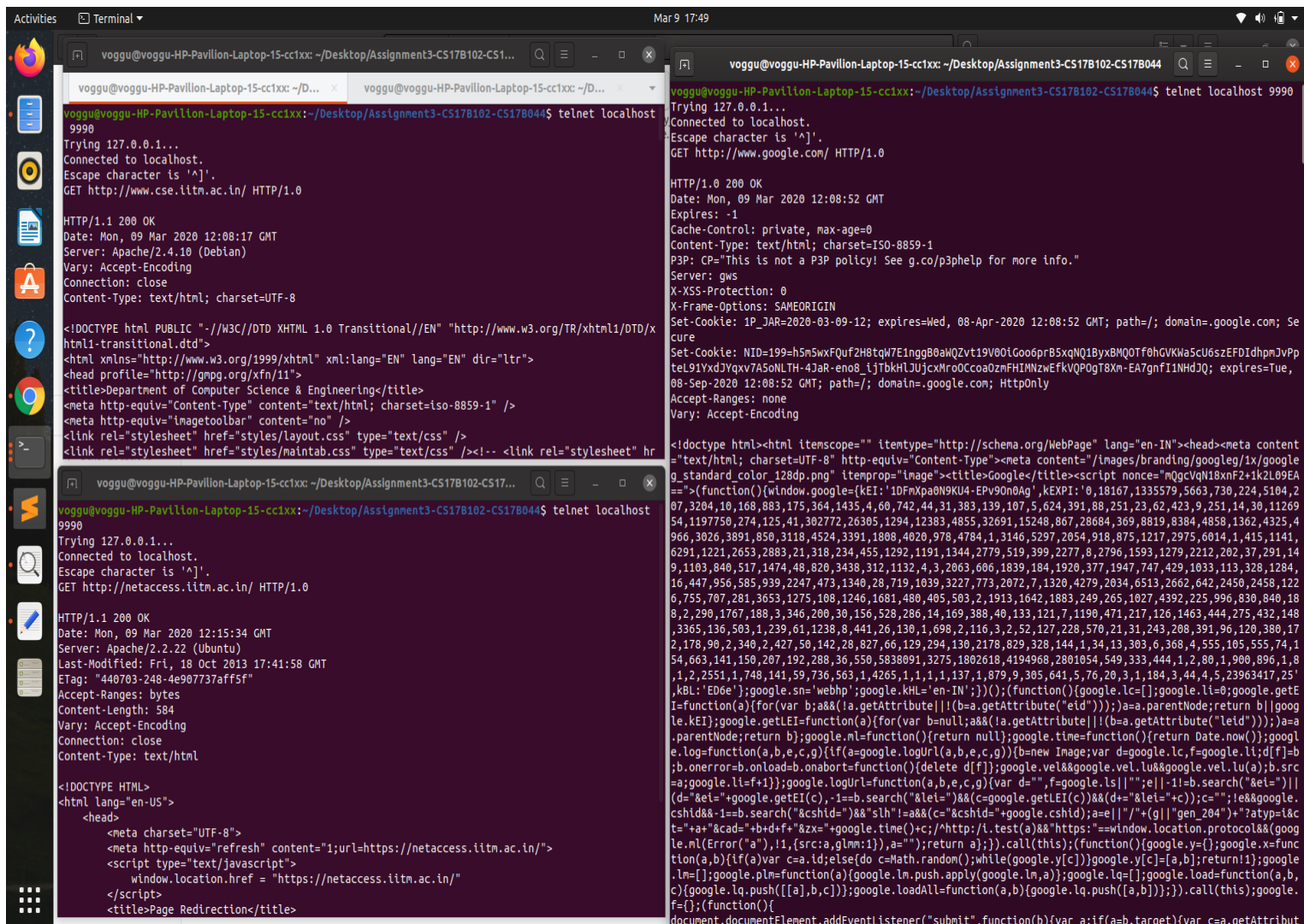
<!-- bxSlider Javascript file <script src="themes/jquery.min.js"></script-->
<script src="themes/jquery.bxslider.min.js"></script>
<!-- bxSlider CSS file -->
<link href="themes/jquery.bxslider.css" rel="stylesheet" />
<!-- sliding images -->

<!-- Piwik -->
<script type="text/javascript">
var _paq = _paq || [];
/* tracker methods like "setCustomDimension" should be called before "trackPageView" */
_paq.push(['trackPageView']);
```

### Analysis:

Our experiment is working fine if we send request from one client as we can clearly see we got reply “200 OK” from receiver end we have used port number as 9999 here.

## Experiment 2: Simultaneous more than one client



The image shows three terminal windows running on a Linux system, each with a different title bar. All three terminals are connected to localhost 9990 via telnet. The first terminal (top left) shows a successful connection to 127.0.0.1 and receives an HTTP 200 OK response from a server running Apache/2.4.10 (Debian). The second terminal (bottom left) shows a similar connection to 127.0.0.1, but the server is Apache/2.2.22 (Ubuntu). The third terminal (right) shows a connection to localhost 9990 and receives a large, complex HTTP response, likely a redirect or a page with a long body. The response includes headers like 'Date: Mon, 09 Mar 2020 12:08:52 GMT' and a body containing HTML and JavaScript code. The terminals are arranged in a way that demonstrates simultaneous connections from multiple clients.

### Analysis:

Our experiment is working fine even if we send requests from more than one client here we have used 3 clients and sent 3 different requests to server and got “OK” replies from each of them in above picture 3 different terminals imply 3 different clients sending requests and they are handled properly.

## Summary:

We've learnt what is socket programming and how it works to a good extent and implemented a model analogous to HTTP proxy server and clearly understood how messages are sent and received at both client and server ends and how proxy manages them. Our proxy acted as client for actual server and server for actual client which mediates the process of sending server request from client to server and received reply from server to client. Basically we have parsed server request from client and sent parsed request to server and collected reply from server and sent it to client using sockets. Mainly we have learnt how HTTP proxy server works and how socket programming works. Experiment was beneficial in learning about HTTP Proxy server works.