Chapter 4

Practical assignment 4

4.1 Background

HTML makes provision for forms that enable one to enter data into fields and then send the data to the server. Consider the following HTML:

```
<form method="get" action="http://www.cs.up.ac.za/">
Number: <input type="text"  name="n" size="20">
<input type="submit" value="Do it">
</form>
```

This will display a form on the screen with a field in which one may enter a number. When one clicks on the 'Do it' button the specified action will be performed; in this case the home page for Computer Science will simply be loaded. As has been explained in practical assignment 3, a `GET` request will be sent to the server concerned (`www.cs.up.ac.za` in this case). Since there is a data field, the content specification (`/` in the `GET` request will be followed by a question mark, which will in turn be followed by the name of the field (`n`), an equals sign and then the value that has been entered by the user. If the user, for example, enters 33 and clicks on 'Do it' the following `GET` request will be sent to the server:

```
GET /?n=33
Host: www.cs.up.ac.za
```

Do your own experiments to see how forms with more than one field work. For more information search the web using your favourite search engine; keep your eyes open for tutorials. Also look again at RFC 2616 if necessary. Those who want to do more, note the `POST` method as an alternative for `GET`.

4.2 Your assignment

Modify the program that you wrote for assignment 2 so that it can be used with a browser (rather than Telnet).

4.3 Assessment

A working program will be awarded 8 out of 10. To earn a higher mark your program has to do more than just the basics - in particular should it demonstrate that you understand something of the HTTP RFC.