Do the following problems from Tanenbaum.

1. Problem 25 on page 717.

Yes, it is certainly possible. The MIME type only tells the type of data on the page. It doesn't specify the helper application to be called. Additionally, Netscape specifically allows you to change the helper application you prefer to use to open your documents.

2. Problem 26 on page 717

There are several ways to interpret it. Here is how I interpreted it: A request comes in every 500 μsec. If it is in cache, it is returned immediately. If it is not, the module has to wait 9 msec for the file to be pulled from disk. In that time, 18 more requests could come in, and if they ALL have to be pulled from disk, the original module will be done processing the disc as soon as the 19th. request arrives, so that it can process this request, meaning 18 modules are sufficient.

Now, this interpretation is based on the understanding that the request can be loaded up while the disk request finishes processing so that the all the module needs to do is check the cache. If we interpret the module has having to actively receive the request, then it won't be able to do this while the disc drive spins, so we will need a 19th. module. My interpretation is based on the idea that since we are told that that the module can magically send information in zero time, it might also be able to magically receive information in zero time, and the 500 μsec just represents the time between requests.

Additionally, we are told that we only have to search the disc half the time, making the average time between disc searches 1 msec. With this in mind, the above reasoning would give us answers of 9 or 10 modules depending on interpretation. I will accept these answers, but they are not as good. Just because disc searches occur half the time does not mean they alternate back and forth in perfect sequence: search, no search, search, no search, etc. If any sequence of 18 requests happens to contain even 10 disc searches, this system fails. Better to use 19 modules to guarantee that the system will work.

3. Problem 27 on page 717.
The actual syntax the web uses is http://www.webpage.com:1234. This will send the request to port 1234.

4. Problem 30 on page 717.

The easiest way would be to store a cookie with an identification code on the client machine. The detailed information would be stored on the server keyed to the client’s identification code.

5. Problem 31 on page 718.

It will work, but it's not secure enough to use for banking. Now, anyone using that account on that machine has access to the banking information. Convenient, yes. Safe, no.