Application Execution Guide

For this practical I used Google Chrome throughout to access and test my webpage and programs.

Upon completion I also tested my webpage with Firefox as per the specification's recommendation and everything worked as expected.

I used additional node.js modules:
Express
fs

1. Open a new terminal and install additional modules by running:
   npm install express
   npm install fs

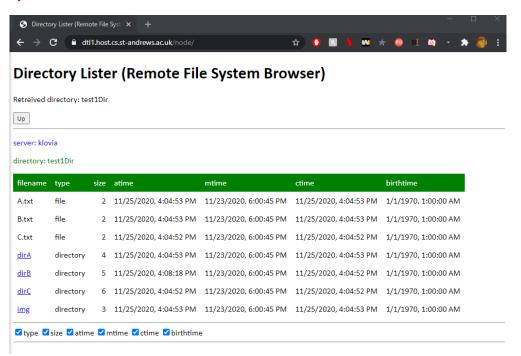2. Navigate to the src folder containing the server

   ```
   dtl1@klovia:~ $ cd Documents/cs2003/P4/src
   dtl1@klovia:~/Documents/cs2003/P4/src $
   ```

3. Supply the name of the directory as a command line argument and run the server by running:
   node dir_list_server.js test1Dir

   ```
   dtl1@klovia:~/Documents/cs2003/P4/src $ node dir_list_server.js test1Dir
   root path: test1Dir
   Listening at port: 21929
   ```

4. Access the directory lister page by opening a web browser (preferably Firefox or Google Chrome) and going to:
   <your_username>.host.cs.st-andrews.ac.uk/node/

   Directory Lister (Remote File System Browser)

   Retreived directory: test1Dir

   [Up]

   server: klovia

   directory: test1Dir

filename	type	size	atime	mtime	ctime	birthtime
A.txt	file	2	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
B.txt	file	2	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
C.txt	file	2	11/25/2020, 4:04:52 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
dirA	directory	4	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
dirB	directory	5	11/25/2020, 4:08:18 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
dirC	directory	6	11/25/2020, 4:04:52 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
img	directory	3	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM

   ☑type ☑size ☑atime ☑mtime ☑ctime ☑birthtime

Design

Application Operation Summary

My application completes requirements 1-7 as listed in the specification. I didn't complete requirement 8 (implementing a search function) as I ran out of time.

My application allows the contents of a remote directory to be viewed, the details of files in the directory are listed in a table.

The user can navigate up and down sub folders in the directory by clicking on them (indicated by their blue colour and underscore).

The user can choose which details of the table are visible via the check boxes below the table and can also sort the listings in ascending or descending order for any of the columns by clicking on the green header for that column.

Additionally, multiple clients are able to browse the same directory simultaneously and independently of each other.

page.html

This file stores the static html skeleton to be served by the server.

I designed the html to best match the screenshot provided in the specification.

I wrap my directory table in a div tag so it can be modified and updated easily by the client.

Under my table I have a row of check boxes that all call the "handleCheck()" method in the client.

page.css

Stylesheet file to style elements in page.html, I styled elements to best match the screenshot provided in the specification.

dir_list_client.js

This is the client-side script file that sends requests to the server, handles the response and updates and modifies elements on the html page.

I use global variables for the client program to store where in the directory the client is and any attributes the user has changed such as hidden columns. I decided to do this rather than have these kinds of variables on the server as this means multiple clients can navigate the directory independently of each other and not interfere.

When deciding how to implement the "up" button I went for an approach to use an array of all directories the client has visited called "dirHistory" I found this useful as I could manipulate the array as needed and make requests for previous directories.

For sorting the listings I made it so clicking the column header once will sort the listings by that column in ascending order, a second click will sort by descending order and a third click will return to the default order as served by the server, I found this to be the most intuitive and least complicated way to achieve this function. I used a global Boolean variable "onDesc" to achieve this.

To sort listings in the table I make use of a simple bubble sort algorithm which I adapted from:
https://stackoverflow.com/questions/7502489/bubble-sort-algorithm-javascript

To sort listings in descending order, I first use this algorithm to sort them in ascending order and then I reverse the order.

My make request function sends a JSON request for directory information for a given directory and then uses the response sent by the server to construct new table rows.
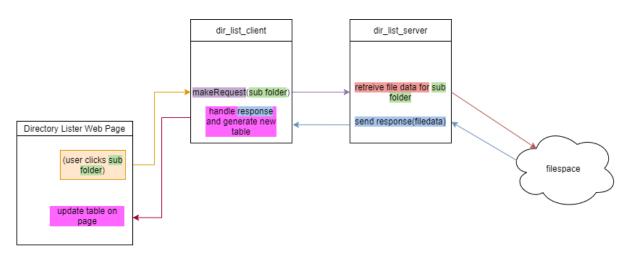
When refreshing / updating the table on the page, I create rows of the table by concatenating string variables and then I update the div "dirTable" on my page with this long html string. I make sure to wrap this new string with the same div so it can be updated in the future.

dir_list_server.js

My server program is fairly simple compared to the client, all my server program does is receive a post request for a directory in the file space, it then makes use of the "getFileInfo" method as supplied on studres which I slightly modified to construct a response to send back to the client.

The server knows nothing about the current client state and at what point in the directory they are at, as explained earlier this is to allow simultaneous clients independent access to the same server.

Operation Diagram

This diagram shows the standard procedure for the communication between the client and the server and how requests and responses are delt with and files accessed.

This diagram, for example, shows what happens when a user wants to navigate to a sub folder.

Testing

Test	Expected Result	Actual Result	Evidence
Not supplying command line argument when starting the server	Error is thrown and console prints ("expected command line argument")	As Expected	
Supplying command line argument and correctly starting server	Server starts	As Expected	
Navigating to https://dtl1.host.cs.st-andrews.ac.uk/node/ in browser	Directory lister page loads and test1Dir is visible	As Expected	

Navigating to sub folder by clicking on one of the links	Table changes to sub folder	As Expected	**Directory Lister (Remote File System Browser)** Retreived directory: test1Dir/dirA [Up] server: klovia directory: test1Dir/dirA | filename	type	size	atime	mtime	ctime	birthtime
A.txt	file	2	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
dirA1	directory	3	11/25/2020, 5:06:41 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM | ☑ type ☑ size ☑ atime ☑ mtime ☑ ctime ☑ birthtime
---	---	---	---
Clicking the "up" button	Table changes to go back to parent folder	As Expected	**Directory Lister (Remote File System Browser)** Retreived directory: test1Dir [Up] server: klovia directory: test1Dir | filename	type	size	atime	mtime	ctime	birthtime
A.txt	file	2	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
B.txt	file	2	11/25/2020, 4:04:53 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
C.txt	file	2	11/25/2020, 4:04:52 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
dirA	directory	4	11/25/2020, 5:06:41 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM
dirB	directory	5	11/25/2020, 4:08:18 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
dirC	directory	6	11/25/2020, 5:06:41 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:52 PM	1/1/1970, 1:00:00 AM
img	directory	3	11/25/2020, 5:06:41 PM	11/23/2020, 6:00:45 PM	11/25/2020, 4:04:53 PM	1/1/1970, 1:00:00 AM | ☑ type ☑ size ☑ atime ☑ mtime ☑ ctime ☑ birthtime

Hiding columns from the table with the checkboxes	Table changes to hide columns	As Expected	
Sorting listings by clicking on "size" column heading	Table sorts by size ascending	As expected	
Clicking on "size" column heading again	Table sorts by size descending	As expected	

Clicking on "size" column heading for 3rd time	Table resets to default sort	As expected	
			<table><thead><tr><th>filename</th><th>type</th><th>size</th><th>atime</th><th>mtime</th><th>ctime</th><th>birthtime</th></tr></thead><tbody><tr><td>A.txt</td><td>file</td><td>2</td><td>11/25/2020, 4:04:53 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:53 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>B.txt</td><td>file</td><td>2</td><td>11/25/2020, 4:04:53 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:53 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>C.txt</td><td>file</td><td>2</td><td>11/25/2020, 4:04:52 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:52 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>dirA</td><td>directory</td><td>4</td><td>11/25/2020, 5:06:41 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:53 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>dirB</td><td>directory</td><td>5</td><td>11/25/2020, 4:08:18 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:52 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>dirC</td><td>directory</td><td>6</td><td>11/25/2020, 5:06:41 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:52 PM</td><td>1/1/1970, 1:00:00 AM</td></tr><tr><td>img</td><td>directory</td><td>3</td><td>11/25/2020, 5:06:41 PM</td><td>11/23/2020, 6:00:45 PM</td><td>11/25/2020, 4:04:53 PM</td><td>1/1/1970, 1:00:00 AM</td></tr></tbody></table>