8.2C: Writing a Server Application

Tasks

This task is to create a server application that displays the **Power 2** table when the user accesses the application via the local web server.

Follow the steps below to complete this task:

1. Create your own local directory (or you can use the same local directory created in Task 8.1).
2. Use express module to create a *Node.js application* (i.e., `expressServer.js`) that will set a local web server. The server listens to the port **8080**. When the user accesses `http://localhost:8080/powertable` using a web browser, the *Power 2* table is displayed.
3. Run the command `node expressServer.js` in a Command Prompt (Windows) or Terminal (Mac OS) within your local directory to make the server work.

4. Open a web browser and visit `http://localhost:8080/powertable`, the *Power 2* table should be displayed in the window like:

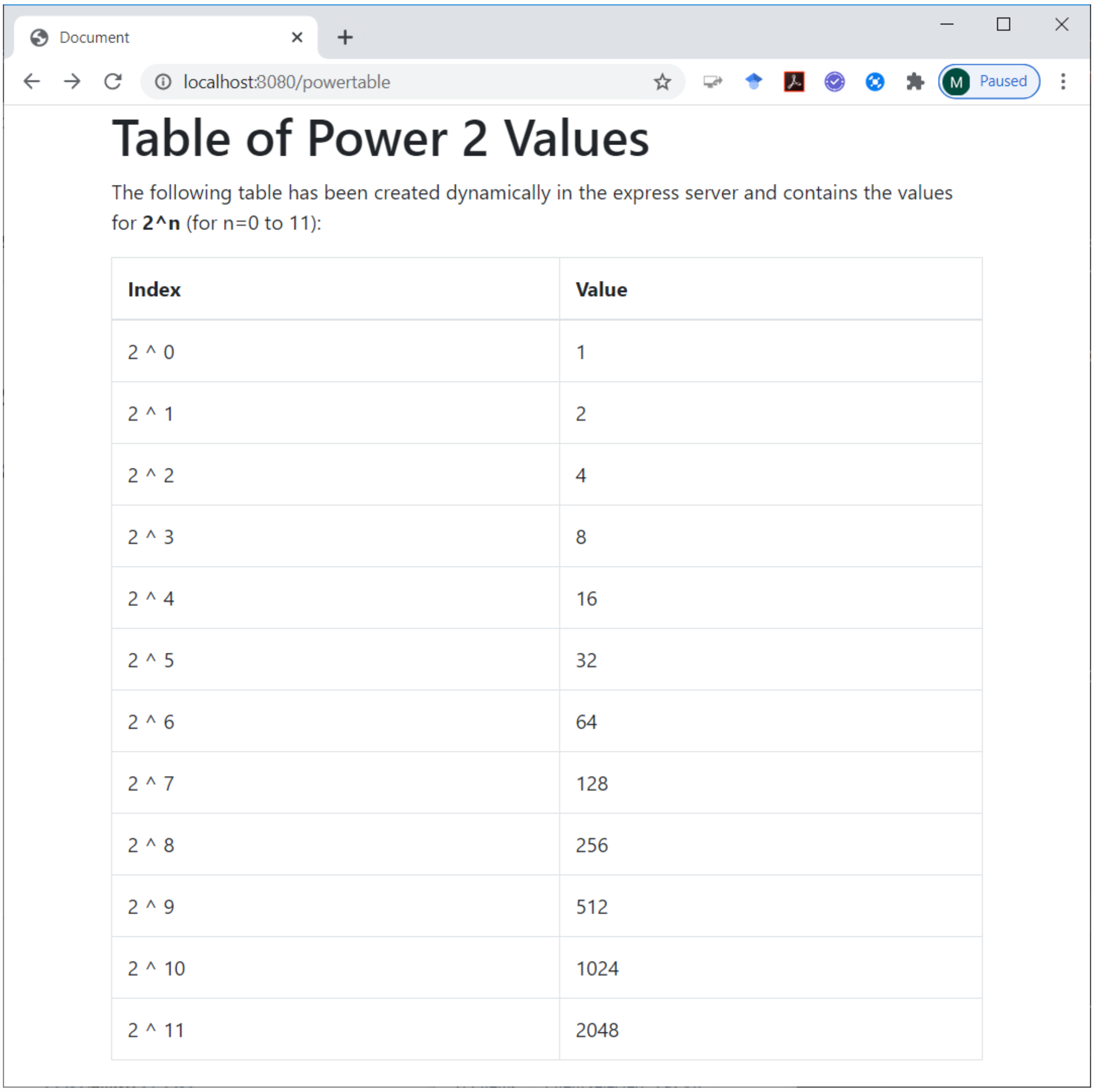

Table of Power 2 Values

The following table has been created dynamically in the express server and contains the values for **2^n** (for n=0 to 11):

Index	Value
2 ^ 0	1
2 ^ 1	2
2 ^ 2	4
2 ^ 3	8
2 ^ 4	16
2 ^ 5	32
2 ^ 6	64
2 ^ 7	128
2 ^ 8	256
2 ^ 9	512
2 ^ 10	1024
2 ^ 11	2048

Task8.2 Node.js application serving generated *Power 2* table page

5. Go back to the Command Prompt or Terminal, and shutdown the server.

Hints

- The structure of the file `expressServer.js` with the *new code* (as highlighted below) could be:

```
// Require the express web application framework (https://expressjs.com)
var express = require('express');

// Create a new web application by calling the express function
var app = express();

// Get port from environment and store in Express.
var port = normalizePort(process.env.PORT || '8080');
app.set('port', port);

// Normalize a port into a number, string, or false.
function normalizePort(val) {
    var port = parseInt(val, 10);
    if (isNaN(port)) {
        // named pipe
        return val;
    }
    if (port >= 0) {
        // port number
        return port;
    }
    return false;
}

// Tell our application to serve all the files under the `public_html` directory
app.use(express.static('public_html'));

// ******NEW CODE******
// This is a 'route' method to handle GET request for /powertable
app.get('/powertable', function(request, response) {
    // Add the JS code here to send a message back to the client with the
    // content of the page to be displayed.
});
// ******NEW CODE******

// Tell our application to listen to requests at port 3000 on the localhost
app.listen(port, function () {
    // When the application starts, print to the console that our app is
    // running at http://localhost:8080  (where the port number is 3000 by
    // default). Print another message indicating how to shut the server down.
    console.log(`Web server running at: http://localhost:${port}`);
    console.log("Type Ctrl+C to shut down the web server");
});
```

- Visit https://expressjs.com/en/guide/writing-middleware.html website to see how to write code for *express functions*.

What will you submit?

You should submit:

- Source code of the file expressServer.js.
- Screenshot of the browser window showing the *Power 2* table when visiting `http://localhost:8080/powertable`.