**WEB700 Assignment 1 - Simple Web Server Simulator**

**Overview**

This JavaScript code serves as a simple web server simulator that processes HTTP requests and returns predefined responses. It consists of three main sections: defining server paths, implementing the web server simulator function (`httpRequest`), and automating tests.

**# Student Information**

- \*\*Student Name:\*\* Rosco Tanner

- \*\*Student Email:\*\* [rdgtanner@myseneca.ca](mailto:rdgtanner@myseneca.ca)

**1. Server Paths**

In this section, the script defines three arrays (`serverVerbs`, `serverPaths`, and `serverResponses`) that store information about HTTP verbs, server paths, and corresponding responses. This information is later used by the web server simulator function.

**2. Web Server Simulator Function - `httpRequest`**

The `httpRequest` function takes two parameters (`httpVerb` and `path`) and simulates a web server's response. It checks if the provided HTTP verb and path match predefined combinations. If a match is found, it returns a response with a status code of 200; otherwise, it returns a 404 error.

# Example Usage

```javascript

console.log(httpRequest("GET", "/")); // Output: "200: Welcome to WEB700 Assignment 1"

console.log(httpRequest("GET", "/about")); // Output: "200: This Assignment was prepared by Rosco Tanner"

console.log(httpRequest("PUT", "/")); // Output: "404: Unable to process PUT request for /"

```

**3. Automating Tests**

The `automateTests` function automates the testing process by continuously invoking the `httpRequest` function with random HTTP verbs and paths. It uses the `setInterval` function to repeat the tests every second.

# Example Output

```bash

404: Unable to process GET request for /logout

404: Unable to process GET request for /randomPath1

200: User Logged In

200: Main Panel

200: User Logged In

...

```

**How to Run**

1. Ensure Node.js is installed on your system.

2. Open a terminal and navigate to the directory containing the script.

3. Run the command: `node assignment1.js`

The script will continuously output simulated web server responses, testing various combinations of HTTP verbs and paths.

A screenshot of a computer program

Description automatically generated