



Delft University of Technology

CSE1500 Web & Database Technology

Final Exam

Date: 27/01/2022

Time Limit: 120 minutes

Instructions:

- This exam contains 22 pages (including this cover page) and 40 multiple-choice questions. Check to see if any pages are missing.
- All questions are worth 1 point.
- The usage of books, notes, old exams, and other written resources is explicitly FORBIDDEN during the exam. The use of electronic aids such as smartphones, laptops, etcetera, is ALSO NOT allowed.
- **There is only one right answer for each question.**
- The order of the answers on your answer form is not always A-B-C-D.
- **There are four possible answers for each question. You can select A, B, C, or D.** It is possible that answer options for a single question are placed cross two pages.
- The questions are not ordered by topic. The ordering was determined so that code blocks running over the end of pages could be avoided.
- Be sure to fill in all header information on the answer form.
- Some questions refer to source code listed a few pages earlier. **Feel free to disassemble your copy of the exam, so that you can work comfortably.**
- This exam copy is yours, you can take it home.

Good Luck!

1 Demo application

A number of exam questions refer to the Web COLOR QUIZ application. It is implemented with Node.js/Express. **We use the short-hand name WCQ from now on.** The goal of WCQ is to familiarize the user with web color names that can be safely used to define colors in CSS. Knowing a handful of color names is usually not a bad idea. Remembering `tomato` is easier than `#ff6347` to define a particular shade of red. The application consists of three screens: the quiz screen (Figure 1), the colors screen (Figure 2) and the information screen (not shown here).

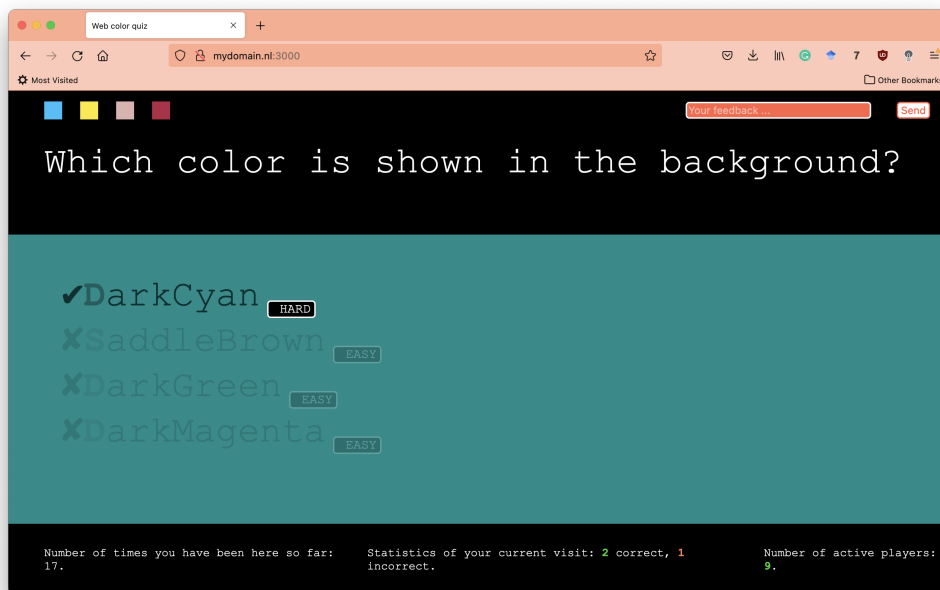


Figure 1: WCQ's quiz screen.

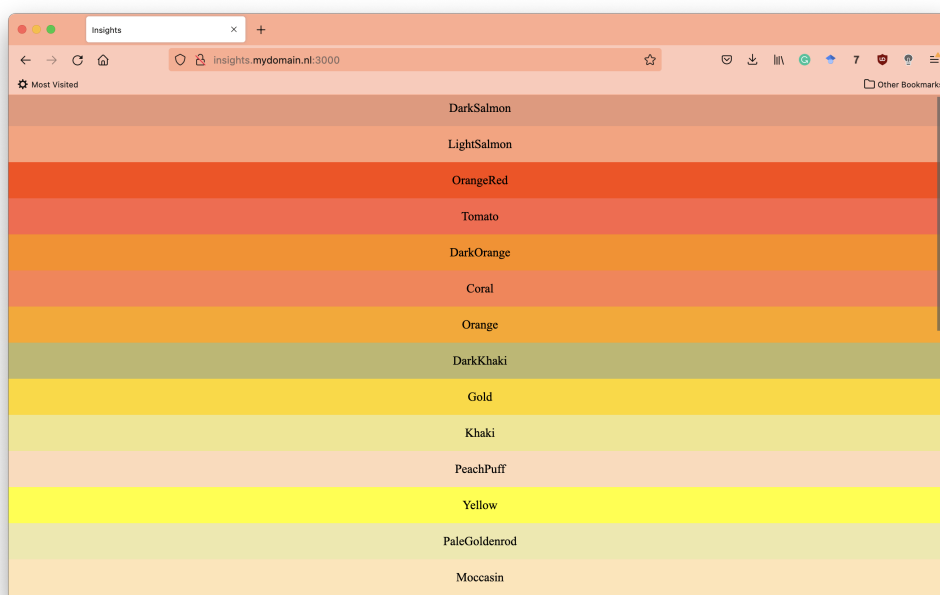


Figure 2: WCQ's colors screen.

The **quiz screen** shows always the same question (*Which color is shown in the background?*) and a changing background color (in Figure 1: **DarkCyan**) whose name the user has to guess. The user is given four color-name choices and selects one. After the selection, the user is shown the correct answer (indicated by a checkmark in front of the correct choice). Two seconds later, a new set of background color and answer options appear and the user can guess again. Each time a quiz question is set up, the client requests four web color names and their difficulty level from the server and randomly selects one color as the correct answer and thus as the color to set the background to. Each color name is associated with a difficulty level, either *easy* (color names that are easy to associate with a color such as **DarkGreen**) or *hard* (e.g. **Orchid**). Each color's difficulty level is revealed when the answer is shown to the user. At the footer of the screen, three statistics are provided: the number of times the user has visited the page (implemented via a cookie), the number of correct and incorrect answers given by the user in the current session and the number of currently active quiz players all over the world (implemented via WebSockets). Finally, at the top right an HTML form allows the user to provide feedback.

The **colors screen** (Figure 2) is for information purposes: it lists all web colors available in the quiz and their name.

The **information screen** contains two pieces of information: the player number assigned to the user and whether or not the user has viewed the colors screen in the past.

Lastly, note that in the questions that follow, questions tagged with **GENERAL** are **NOT** about the WCQ application. Questions tagged with **WCQ** are referring to the WCQ application.

2 Questions

Question 1

GENERAL Consider the following HTTP header:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 2465
Date: Tue, 07 Dec 2021 22:33:00 GMT
```

Which of the following statements about it is TRUE?

- A. It is a valid HTTP request header.
- B. It is a valid HTTP response header.**
- C. It is an invalid HTTP request header, as HTTP requests require status codes above 400.
- D. It is an invalid HTTP response header, as the required **Set-Cookie** attribute is missing.

Question 2

GENERAL Which of the following statements about HTTP/1.1 is FALSE?

- A. HTTP is an application-level protocol.
- B. HTTP messages consist of requests from client to server and responses from server to client.
- C. Every HTTP message must have a message body.**
- D. The request-header fields allow the client to pass additional information about the request, and about the client itself, to the server.

Question 3

WCQ In WCQ we find a feedback form at the top right (Figure 1). This feedback form should allow a user to send feedback about the app to the app creators. Inspecting the source code for the feedback form, we find the piece of HTML below. Which of the following statements about it is TRUE?

```
1 <form id="feedbackForm" action="/sendFeedback" method="get">
2   <input type="hidden" id="pathname" name="pathname" value="">
3   <input type="text" id="feedback" name="feedback" placeholder="Your feedback">
4   <input type="button" id="submit" value="Send">
5 </form>
```

- A. The feedback form will not work as envisioned. The `action` attribute contains a path to a local file, instead of an absolute URL to send the data to. No data is sent to the server.
- B. The feedback form will not work as envisioned. The `method` attribute is set to `get` which means the form is set up to *receive* data from the server instead of *sending* data to the server.
- C. The feedback form will not work as envisioned. If one or more `<input>` elements have the type `hidden`, the entire `<form>` is hidden from the server. No data is sent to the server.
- D. The feedback form will work as envisioned if the server defines a route `/sendFeedback`. The values of the `<input>` elements with names `feedback` and `pathname` will be sent to the server.**

Question 4

WCQ Imagine you are asked to work on the development of WCQ's client-side code. You set up the project in Visual Studio Code, start the Node.js script (let us assume it listens on port 3000) and make changes to some of the HTML and CSS files. Each time you save those changes, you reload the browser tab pointing to `localhost:3000` to see the effect of the changes you made. This gets tiresome quickly and you decide to write a piece of client-side JavaScript. What can you do to efficiently (using the least bandwidth possible) and automatically reload `localhost:3000` after changes have been made to the HTML and CSS files?

- A. This is not possible. JavaScript cannot trigger the reload of a browser tab.
- B. Set up a piece of JavaScript that approximately once a second makes HTTP requests using the `UPDATE` method for all client-side files. The server will only return files that have changed since the last HTTP request made by the client (and otherwise status code `201:No Update` is returned). If a file is retrieved this way, the script triggers a reload of the browser tab with all files reloaded from the local browser cache.
- C. Set up a piece of JavaScript that once a second makes HTTP requests using the `HEAD` method for all client-side files. If a change in any returned `HEADER` is detected compared to the previous version, the script triggers a reload of the browser tab.**
- D. Set up a piece of JavaScript that once a second triggers a reload of the browser tab.

Question 5

GENERAL How do Web caches employ the `If-Modified-Since` HTTP header field to work efficiently?

- A. A Web cache only sends an HTTP request to an origin server when the `Cache-Control` and the `If-Modified-Since` dates coincide.
- B. A Web cache sends an HTTP GET request to an origin server with the `If-Modified-Since` header field set. In reply to this request, the origin server sends an HTTP response. This response only contains a content body if the resource has changed after the date found in `If-Modified-Since`.**
- C. A Web cache only sends an HTTP request to an origin server if the Web resource has been modified by the origin server in the meantime.
- D. A Web cache sends an HTTP GET request to an origin server. In reply to this request, the origin server sends an HTTP response. If this response contains an `If-Modified-Since` header field with a value set to a date in the past, the Web cache does not store the response.

Question 6

GENERAL Which of the following statements describes long polling most accurately?

- A. After the initial HTTP response, the server can send HTTP responses directly to the client when new content is available without having to wait for another HTTP request. The client renders the updated content.
- B. The client's first HTTP request contains information about the WebSocket protocol update that is required for the server to send HTTP responses to the client without having to wait for another HTTP request.
- C. After the initial HTTP response, the client sends an HTTP request and the server holds the request open until new content is available before sending its HTTP response. Once the response is sent, the client immediately sends another HTTP request that is kept open.**
- D. After the initial HTTP response, the client regularly sends an HTTP request to the server, the server in turn sends its HTTP response and if the response contains different content than before the client renders the updated content.

Question 7

GENERAL Consider a web application which on the server-side logs all HTTP requests. On the server-side, the application does neither require login information from the users nor does it personalize HTTP responses. Now, based on these logs, logical sessions should be constructed, that is, to identify series of HTTP requests that were coming *most likely* from the same user. What approach or HTTP request header would be most appropriate to use in this scenario?

- A. The unique IDs present in the path of each (fat) URL
- B. `WWW-Authenticate` header field information
- C. `User-Agent` header field information**
- D. `Authorization` header field information

Question 8

GENERAL Consider the following four statements about URLs:

- (1) When a URL starting with scheme `http` is entered into the browser's address bar, the browser uses port 80 by default.

- (2) When entering the URL `https://github.com/chauff/IN4325#course-desc` in the browser's address bar and pressing [Enter], the server sends only a part of the resource residing at path IN4325 (namely the part that starts at local anchor `course-desc`).
- (3) The URL `https://duckduckgo.com/?q=chess&t=h_&ia=web` contains an empty path and several query components.
- (4) URLs with Unicode characters can be converted reversibly and uniquely to an ASCII string with Punycode encoding.

Which of these statements are TRUE?

- A. Only 1) and 2) and 3)
- B. Only 1) and 3) and 4)**
- C. Only 2) and 4)
- D. All four statements are true.

Question 9

GENERAL Consider the following telnet exchange on the terminal. Which of the following statements about it is TRUE?

```
bash-3.2$ telnet microsoft.com
Trying 104.215.148.63...
Connected to microsoft.com.
Escape character is '^]'.
HEAD / HTTP/1.1
host:microsoft.com:80

HTTP/1.1 301 Moved Permanently
Date: Wed, 08 Dec 2021 08:55:59 GMT
Server: Kestrel
Location: https://www.microsoft.com/

Connection closed by foreign host.
```

- A. The IP address shown here is not a valid IPv4 address.
- B. The Domain Name System server resolution step is missing in this telnet exchange.
- C. The initial telnet command is invalid as the port number is missing.**
- D. The Location header field is invalid as it should contain an IP address instead of a URL.

Question 10

GENERAL Consider the URL below:

```
ftp://anonymous:mypass@ftp.csx.cam.ac.uk/gnu;date=today
```

What URL component is not present in this URL?

- A. <scheme>
- B. <path>
- C. <query>**
- D. <params>

Question 11

GENERAL Which of the following statements about HTTP basic authentication is FALSE?

- A. The basic authentication scheme transmits credentials as user-id/password pairs, encoded using base-64.
- B. Basic authentication is not a secure method of user authentication unless used in conjunction with TLS (or another equivalent secure system).
- C. Upon receipt of a request for a URL within the protection space that lacks credentials, the server can reply with a challenge using the 401 (Unauthorized) status code.
- D. A client must not preemptively send the corresponding Authorization header field with requests for resources in the protection space. The client always has to wait for a challenge from the server.**

Question 12

WCQ A performance analysis of WCQ reveals that its CLS score is too large (and large means that WCQ performs poorly). Which of the following strategies is useful to decrease the application's CLS score?

- A. Keep the number of HTTP requests required to load the entire application as low as possible.
- B. For applications that include image and/or video elements, reserve the required space on the page for each image/video before it is retrieved by the browser with a HTTP request.**
- C. Split up long-running JavaScript code into smaller tasks.
- D. When inserting content in a web application in response to user interactions, insert it above the content currently visible in the viewport.

Question 13

GENERAL Which of the following statements about the HTML section heading elements is FALSE?

- A. A web page should only contain a single `<h1>` element.
- B. There are six heading elements in total, `<h1>` to `<h6>`.
- C. The content inside an h1 element cannot be longer than what can be rendered within a single line inside the viewport.**
- D. The section heading elements should be used in the order of their numbering; for example, if a sub-section header is represented by an `<h4>` element, a sub-sub-section header should not be represented by an `<h3>` element.

Question 14

GENERAL Which of the following statements about the CSS `position` property is TRUE?

- A. When an element has value `relative`, the element is adjusted on the fly, all other elements are affected too.
- B. When an element has value `fixed`, the element is positioned relative to the viewport and remains in that position when scrolling.**
- C. When an element has value `absolute`, the element is taken out of the normal flow but space remains reserved for it.
- D. When an element has value `static`, the element is taken out of the normal flow and no space remains reserved for it.

Question 15

GENERAL When executing the following piece of JavaScript in the browser's web console, what will be the output on the console?

```
1 function changeColorShade(amount){
2   return function(rgb){
3     return [Math.min(rgb[0]*amount, 255),
4             Math.min(rgb[1]*amount, 255),
5             Math.min(rgb[2]*amount, 255)]
6   }
7 }
8 let lighter = changeColorShade(1.2);
9 let darker = changeColorShade(0.8);
10
11 console.log(lighter([200,10,60]))
12 console.log(darker([200,10,60]))
```

- A. The output on the console is as seen below. This is due to the scope of `amount` not being resolved: in the inner function call the argument `amount` is `undefined`.
- ```
[NaN, NaN, NaN]
[NaN, NaN, NaN]
```
- B. The output on the console is as seen below. This is due to the `rgb` parameter being expected to be of type `number` instead of type `object`.
- ```
[ NaN, NaN, NaN ]
[ NaN, NaN, NaN ]
```
- C. The output on the console is:**
- ```
[240, 12, 72]
[160, 8, 48]
```
- D. The output on the console is:
- ```
[ 200, 10, 60 ]
[ 200, 10, 60 ]
```

Question 16

GENERAL Which of the following statements about the WebSocket protocol is FALSE?

- A. The WebSocket protocol enables peer-to-peer communication between two browsers, i.e. two clients can communicate directly with each other, without the need for a server.**
- B. Clients can initiate WebSocket connections with servers by sending a HTTP request with a `Connection:Upgrade` header field.
- C. Servers cannot initiate a WebSocket connection.
- D. The WebSocket protocol enables bidirectional communication, an improvement over HTTP 1.1 which cannot be used for bidirectional communication.

Question 17

GENERAL Which of the following manipulations does NOT enable an injection attack?

- A. DOM manipulation**
- B. URL parameter manipulation
- C. HTTP header manipulation
- D. Hidden form field manipulation

Question 18

GENERAL What will be the result of executing this piece of JavaScript in the browser?

```
1 window.used = false
2
3 function Color(n, m, u){
4     this.name = n;
5     this.mood = m;
6     this.used = u;
7
8     this["isInUse"] = function(){
9         return used;
10    }
11 }
12
13 let c = new Color("tomato", "happy", false);
14
15 Color.prototype.hasBeenUsed = function(){
16     used = true;
17 }
18
19 console.log(c.isInUse())
20 c.hasBeenUsed()
21 console.log(c.isInUse())
```

- A. true
true
- B. false
false
- C. true
false
- D. false
true**

Question 19

GENERAL Which of the following statements about the middleware functions of Node.js is FALSE?

- A. Middleware functions are functions that have access to the request object, the response object, and the next middleware function (variable named `next`) in the application's request-response cycle.
- B. If the current middleware function does not end the request-response cycle, it must call `next()` to pass control to the next middleware function. Otherwise, the request will be left hanging.
- C. Middleware components cannot change the request and response objects.**
- D. Middleware functions can execute code.

Question 20

GENERAL Which of the following statements about cookies is TRUE?

- A. Secure cookies allow the server to check whether the cookie value has been tampered with by the client.
- B. Transient cookies are also called session cookies and only exist in the browser's local storage.
- C. If a cookie has no explicit expiration date, it is a persistent cookie.
- D. Third-party cookies are cookies that belong to domains different from the one shown in the browser's address bar.**

Question 21

GENERAL Which of the statements about the following piece of code is TRUE?

```
1 function Color(n) {
2   this.name = n;
3 }
4
5 Color.prototype.getName = function () {
6   return this.name;
7 };
8
9 function ColorMood(n, m) {
10  Color.call(this, n);
11  this.mood = m;
12 }
13
14 ColorMood.prototype.constructor = ColorMood;
15
16 ColorMood.prototype.getMood = function () {
17   return this.mood;
18 };
19
20 function ColorMoodUsage(n, m, u) {
21  ColorMood.call(this, n, m);
22  this.usage = u;
23 }
24
25 ColorMoodUsage.prototype.constructor = ColorMoodUsage;
26
27 let c1 = new Color("seagreen");
28 let c2 = new ColorMood("gold", "happy");
29 let c3 = new ColorMoodUsage("tomato", "friendly", 0);
30
31 ColorMoodUsage.prototype.incrUsage = function () {
32   return ++this.usage;
33 };
34
35 console.log(c3.getName());
36 console.log(c3.getMood());
37 console.log(c3.incrUsage());
```

- A. Object c3 has a direct link to two prototype objects.
- B. Object c3 tries to access properties (getName, getMood) that the object does not have. Due to a broken prototype chain, no matching properties are found in the chain.**
- C. Property incrUsage was added to the prototype after the creation of object c3 and thus c3 cannot access this property.
- D. As the properties name and mood have been defined as private properties of c1 and c2 respectively, they are not accessible to object c3.

Question 22

GENERAL Consider the following four statements about Ajax:

- (1) Ajax is a Node.js mechanism that enables the dynamic loading of content without having to reload the page manually.
- (2) Ajax revolves around the use of the window object to communicate between client and servers.
- (3) Using Ajax, servers can send several HTTP responses in reply to a single HTTP request.
- (4) Ajax reduces the HTTP request/response cycle overhead as a complete HTTP message is no longer required for the communication between client and server.

Which of these statements are TRUE?

- A. Only 1) and 2)
- B. Only 3) and 4)
- C. All four statements are true.
- D. None. All four statements are false.**

Question 23

WCQ Consider the following function, which is the registered event handler for the `click` event on any of the four `` elements which represent a possible answer (i.e. one of the colors) on the quiz screen.

```
1 function answerClick(e) {
2
3 //there is only one <ul> element on the quiz screen
4 const children = document.getElementsByTagName("ul")[0].children;
5
6 for (let i = 0; i < children.length; i++) {
7   const li = children[i];
8
9   /**
10    * CSS class selector 'visible': shows checkmark and cross-mark depending
11    * on whether the answer is correct or incorrect
12    */
13   li.classList.add("visible");
14 }
15
16 /**
17  * setQuestion is a function that retrieves a new set of
18  * colors from the server to guess.
19  */
20  setTimeout(setQuestion, 3000);
21 }
```

What happens if a user clicks **3** times on one of the answers within a **1 second** time window?

- A. After the first click of the user, the answer correctness is made visible to the user. After approximately three seconds measured from the first click three new questions are shown to the user in quick succession. Nothing else happens.**
- B. After the first click of the user, the answer correctness is made visible to her. After approximately 3 seconds measured from the first click, a new question is shown. Nothing else happens.
- C. After the third click of the user, the answer correctness is made visible to her. After approximately three seconds measured from the last click, a new question is shown. Nothing else happens.
- D. After the third click of the user, the answer correctness is made visible to the user. After 30 seconds measured from the first click, a new question is shown. Nothing else happens.

Question 24

GENERAL Web portals may secure themselves against CSRF attacks using CSRF tokens. Those tokens often appear as hidden fields in HTML forms. Below you can find four possible characteristics of CSRF tokens:

- (1) The token should have the `Secure` flag set.
- (2) The token value should be the base64-encoding of the session ID.
- (3) The token should be verifiable on the server-side.
- (4) The token should be unique per user session and predictable, so the client can verify its authenticity.

Which of these characteristics should a good CSRF token (sent in a hidden form field) have?

- A. Only 1) and 4)
- B. Only 3)**
- C. Only 2)
- D. All of them.

Question 25

GENERAL What does the user see when opening the web page whose source code is shown below in the browser and moving the mouse across the page?

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <style>
6     body {
7       height: 100vh;
8       width: 100vw;
9     }
10  </style>
11  <script>
12    const startColor = [120, 0, 255];
13    const endColor = [0, 255, 120];
14
15    const body = document.createElement("body");
16    body.addEventListener('mousemove', function(e) {
17      const x = e.clientX / window.innerWidth;
18
19      const array = [
20        Math.round((endColor[0] - startColor[0]) * x + startColor[0]),
21        Math.round((endColor[1] - startColor[1]) * x + startColor[1]),
22        Math.round((endColor[2] - startColor[2]) * x + startColor[2])
23      ];
24      body.style.backgroundColor = "rgb(" + array.join(',') + ")";
25    });
26  </script>
27 </head>
28 <body></body>
29 </html>

```

- A. A white background; no error is shown on the web console.**
- B. A background whose color changes from violet to green, depending on the location of the mouse pointer.
- C. A white background; an error is printed on the console: Uncaught TypeError: body is undefined.
- D. A white background; an error is printed on the console: Uncaught TypeError: body.addEventListener is not a function.

Question 26

GENERAL Given two files `bar.js` and `foo.js`, what is the output on the console after running `node foo.js`?

```

1 module.exports = function () { //bar.js
2   return [
3     function () {
4       return "gold";
5     },
6     function () {
7       return "silver";
8     },
9     function () {
10      return "bronze";
11    },
12  ];
13 };

```

```
1  const medalsOld = require("./bar.js")(); //foo.js
2  console.log(medalsOld[0]());
3
4  medalsOld.push(function () {
5    return "copper";
6  });
7
8  const medalsNew = require("./bar.js")();
9  console.log(medalsNew[3]());
```

- A. gold
copper
- B. TypeError: medalsNew[3] is not a function**
- C. [Function]
[Function]
- D. gold
bronze

Question 27

WCQ The list of web colors that are served during this quiz are stored on the server in a file called `webcolor.json`. The content of the file looks as follows (note that we here show three colors, but the actual file contains many more):

```
1  [
2    { "name": "MediumVioletRed", "difficulty": "easy" },
3    { "name": "DeepPink", "difficulty": "easy" },
4    { "name": "DarkSeaGreen", "difficulty": "hard" }
5  ]
```

One of the quiz developers wrote the following Node.js script to help all developers to keep an eye on the number of colors in `webcolor.json` while they are at home. The script is stored in a file called `colors.js` and started as follows on the server (which has IP address 8.8.8.8): `node colors.js 4001`. The developers in turn connect to the server via telnet using the following command: `telnet 8.8.8.8 4001`. The idea is for them to receive the number of colors in `webcolor.json` approximately every five seconds as a printout on the terminal. What actually happens instead?

```
1  const fs = require("fs");
2  const net = require("net");
3
4  const file = "webcolors.json";
5  const port = process.argv[2];
6
7  const server = net.createServer(function (connection) {
8    setTimeout(function () {
9      fs.readFile(file, function (err, data) {
10         const entries = JSON.parse(data);
11         connection.write(`Number of colors in the system is ${entries.length}\n`);
12       });
13     }, 5000);
14
15     connection.on("close", function () {});
16   });
17
18   server.listen(port, function () {
19     console.log("Waiting for clients to connect.");
20   });
```

Note: assume that the file `webcolors.json` contains exactly three colors.

- A. **Approximately five seconds after connecting, a single message is printed on the terminal** `Number of colors in the system is 3.`; afterwards no further messages are printed. This behavior is the same for all clients that connect to the server via telnet.
- B. No message is printed on the terminal. This behavior is the same for all clients that connect to the server via telnet.
- C. Approximately every five seconds, the message `Number of colors in the system is 0.` is printed on the terminal. This behavior is the same for all clients that connect to the server via telnet.
- D. The behavior differs depending on when a client connects to the server. The first client that connects receives the message `Number of colors in the system is 0.` approximately every five seconds. Clients that connect later receive no message.

Question 28

WCQ When looking at the Node.js script of WCQ, the last route in the file is defined as follows:

```
1 app.use("*", function (req, res) {
2   res.writeHead(404, {
3     "Content-Type": "text/plain; charset=utf-8",
4     "Content-Length": "500",
5   });
6   res.end("Error: this route is not implemented.");
7 });
```

Consider the following statements about the Node.js file system:

- (1) A file is its own module.
- (2) A file accesses its own module definition through the `_self` variable.
- (3) Modules are cached after they have been loaded N times in an application, where N is a threshold fixed by the runtime environment.
- (4) Node.js has several core modules compiled into the binary which load automatically when Node.js starts.

Which of the these statements are FALSE?

- A. Only 1) and 3) and 4)
- B. Only 1) and 2)
- C. Only 1) and 4)
- D. Only 2) and 3)**

Question 29

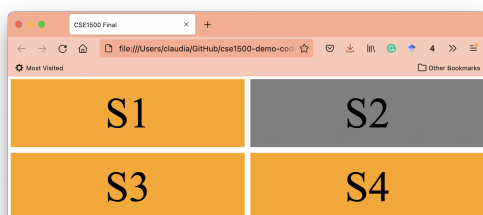
GENERAL Consider the following HTML:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>CSE1500 Final</title>
6   <style>
7     * {
8       margin: 0;
9     }
10
11     #main {
12       display: grid;
13       grid-template-columns: 1fr 1fr;
14       grid-template-rows: 1fr 1fr;
15     }
16
17     .square {
18       position: relative;
19       width: calc(50vw - 10px);
20       height: calc(50vh - 10px);
21       background-color: lightblue;
22       color: black;
23       border: 5px solid white;
24       font-size: 75px;
25       display: flex;
26       align-items: center;
27       justify-content: center;
28     }
29
30     .square + .square {
31       background-color: grey;
32     }
33
34     div > .square {
35       background-color: orange;
36     }
37
38     .square + .square + .square {
39       background-color: brown;
40     }
41   </style>
42 </head>
43
44 <body>
45   <div id="main">
46     <div class="square odd">S1</div>
47     <div class="square even">S2</div>
48     <div class="square odd">S3</div>
49     <div class="square even">S4</div>
50   </div>
51 </body>
52
53 </html>

```

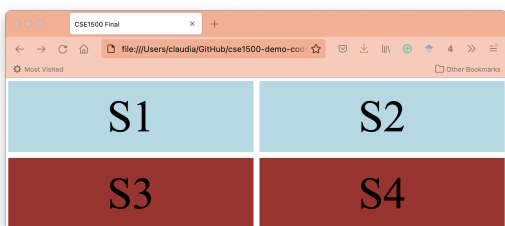
How will its rendering look like?



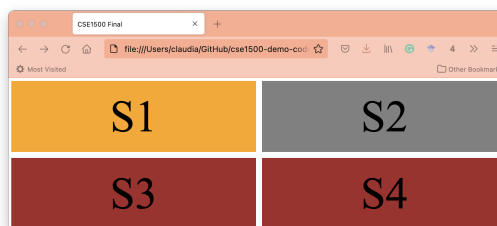
Q29: Answer A.



Q29: Answer B.



Q29: Answer C.



Q29: Answer D.

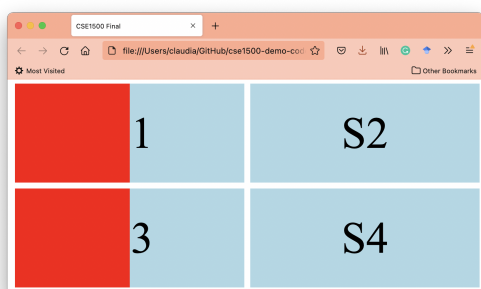
Question 30

GENERAL Consider the HTML of Question 29 once more. Replace lines 30-40 of the source with the following code snippet:

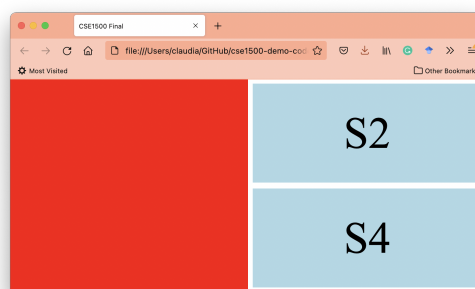
```

1  .square::after {
2      content: " ";
3      position: absolute;
4      top: 0;
5      left: 0;
6      width: 50%;
7      height: 100%;
8      background-color: red;
9  }
    
```

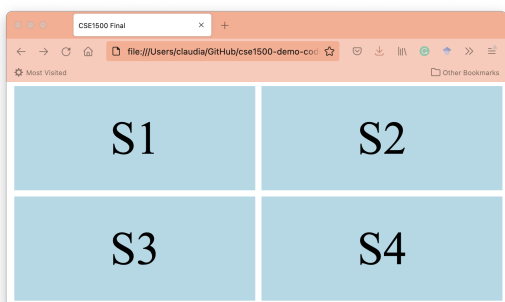
How will the rendering look like now?



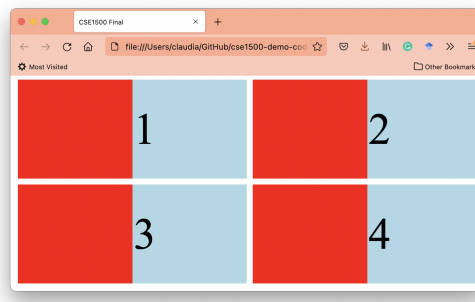
Q30: Answer A.



Q30: Answer B.



Q30: Answer C.



Q30: Answer D.

Question 31

GENERAL Consider the HTML of Question 29 one last time. Which of the following selectors does **NOT** select the two `<div>` elements whose inner HTML is set to `S1` and `S3` respectively?

- A. `.square.odd { /* ... */ }`
- B. `div:nth-of-type(2n+1) { /* ... */ }`
- C. `div:nth-child(2n+1) { /* ... */ }`
- D. `.square .odd { /* ... */ }`

Question 32

WCQ In WCQ, at the top left four decorative squares are visible: they are animated and change their color every half a second. Each square is a `<div>` element with class `square`. Their styling is defined by the following CSS snippet:

```
1 .square {
2   position: absolute;
3   width: 25px;
4   height: 25px;
5   top: 15px;
6   left: 50px;
7   background-color: tomato;
8   animation: colors 10s infinite;
9 }
10 @keyframes colors {
11   0% {
12     background-color: tomato;
13   }
14   5% {
15     background-color: deepskyblue;
16   }
17   10% {
18     background-color: gold;
19   }
20   /* .. more colors in 5% increments */
21   95% {
22     background-color: seagreen;
23   }
24 }
```

Based on this snippet, all four squares change their background color at the same time. Which CSS property do we need to add to every single square to make the color changes appear out of sync?

- A. `animation-transform`
- B. `data-attribute`
- C. `timing-delay`
- D. **None of the above properties are valid for this purpose.**

Question 33

Consider the following four statements describing roles of the OAuth 2.0 authorization framework:

- (1) The resource server hosts the protected resources, and is capable of accepting and responding to protected resource requests using access tokens.
- (2) The client is the entity that grants access to a protected resource.
- (3) The authorization server issues access tokens to the client after successfully authenticating the resource owner and obtaining authorization.
- (4) The resource owner makes protected resource requests on behalf of the client and with its authorization.

Which of these role descriptions are correct?

- A. Only 1) and 3)
- B. Only 2) and 4)
- C. All of them.
- D. None of them.

Question 34

WCQ Consider the following excerpt of the quiz screen's HTML:

```
1 <body>
2   <header> Which color is shown in the background?
3     <div class="square"></div>
4     <div class="square"></div>
5     <div class="square"></div>
6     <div class="square"></div>
7   </header>
8   <main>
9     <ul id="answers"></ul>
10  </main>
11  <footer>
12    <!-- footer elements -->
13  </footer>
14 </body>
```

To make the question text inside the `<header>` tag look visually more appealing, you want to assign different colors to each letter of the word `color`, e.g. `color`. Is this possible in CSS without additional JavaScript code or making changes to the HTML excerpt?

- A. Yes, using pseudo-class `nth-of-type`.
- B. Yes, using pseudo-elements `first-letter`, `second-letter` and so on.
- C. Yes, using pseudo-class `nth-child`.
- D. No, this is not possible in CSS.

Question 35

WCQ The information screen of WCQ contains information about the user (such as the user ID). The following route handler has been defined for this screen:

```
1 app.get("/w(ho)?(hat)?am+i*", function (req, res) {
2   res.render("information.ejs");
3 });
```

Assume that the server is started on the local machine. Now consider these four URLs:

- `http://localhost:3000/whoamiii`
- `http://localhost:3000/whatai`
- `http://localhost:3000/whatamidoinghere`
- `http://localhost:3000/whowhatami`

How many of them match the route defined above?

- A. 1
- B. 2
- C. 3
- D. 4

Question 36

WCQ The WCQ app allows the player to send feedback. The following Node.js script shows how this feedback is processed:

```
1  const express = require("express");
2  const url = require("url");
3  const http = require("http");
4
5  const app = express();
6  const port = 3333;
7
8  http.createServer(app).listen(port);
9
10 function parseFeedback(u) {
11   const parsed = url.parse(u, true).query;
12   const feedback = parsed["feedback"] !== undefined ? parsed["feedback"] : "n/a";
13   return feedback;
14 }
15
16 app.get(
17   "/feedback",
18   function (req, res, next) {
19     const feedback = parseFeedback(req.url);
20     if (feedback.includes("bad") || feedback.includes("terrible")) {
21       next();
22     } else {
23       res.send("Why did you have to say this?");
24     }
25   },
26   function (req, res, next) {
27     const feedback = parseFeedback(req.url);
28     if (feedback.includes("ok") || feedback.includes("not too bad")) {
29       next();
30     } else {
31       res.send("Really?");
32     }
33   },
34   function (req, res, next) {
35     const feedback = parseFeedback(req.url);
36     if (feedback.includes("good") || feedback.includes("great")) {
37       next();
38     } else {
39       res.send("Wow!");
40     }
41   }
42 );
43
44 app.get("/feedback", function (req, res, next) {
45   res.send("Thank you for your feedback!");
46 });
```

The script is started on the local machine. These four URLs (in this order) are accessed:

`http://localhost:3333/feedback?feedback=great%20game`

`http://localhost:3333/feedback?feedback=not%20too%20bad`

`http://localhost:3333/feedback?feedback=this%20is%20bad`

`http://localhost:3333/feedback?feedback=ok%20game,%20bad%20sound,%20good%20visuals`

Which of the following responses will you see in the browser (in this order)?

A. Why did you have to say this?

Wow!

Really

Thank you for your feedback

- B. Thank you for your feedback
Thank you for your feedback
Thank you for your feedback
Thank you for your feedback
- C. Wow!
Really?
Why did you have to say this?
Thank you for your feedback!

D. None of the above

Note: while answer (A) was meant to be the correct answer, due to the missing punctuation, we give the point for both answers (A) and (D).

Question 37

WCQ The WCQ app maintains information on the number of currently active players as seen at the bottom of Figure 1. Below is a condensed version of the Node.js script that is responsible for determining the number of currently active players:

```
1  const express = require("express");
2  const http = require("http");
3  const cookies = require("cookie-parser");
4  const sessions = require("express-session");
5  const credentials = require("./credentials.js");
6  const vhost = require("vhost");
7
8  const port = 3333;
9  let playerNumber = 0;
10
11 const app = express();
12 app.use(express.urlencoded({ extended: true }));
13 app.use(express.json());
14 app.use(cookies(credentials.cookieSecret));
15 const sessionConfiguration = {
16   secret: credentials.cookieSecret,
17   resave: false,
18   saveUninitialized: true,
19 };
20 app.use(sessions(sessionConfiguration));
21
22 app.get("/getActivePlayers", function (req, res) {
23   if (!req.cookies.playerNumber) {
24     playerNumber++;
25     res.cookie("playerNumber", playerNumber, {
26       path: "/",
27       maxAge: 1000 * 60 * 60 * 24 * 365,
28     });
29   }
30
31   const session = req.session;
32   if (session.views) { session.views++;}
33   else {session.views = 1;}
34
35   /*
36    * req.sessionStore.all is a function that in a callback
37    * returns all the sessions in the store. Each session
38    * is one object property (id that identifies the session).
39    * Object.keys() returns an array of the object's properties.
40    */
41   req.sessionStore.all(function (err, sessions) {
42     if (!err) {
43       es.send(Object.keys(sessions).length + " active players");
44     } else {
45       res.send("Number of active players: 0");
46     }
47   });
48 });
49
50 http.createServer(app).listen(port);
```

This script is started on the local machine and a user now executes the following actions (on the same machine, within a 1-minute time window):

1. The user opens the Firefox browser and accesses `http://localhost:3333/`
2. The user opens a new tab in the same instance of the Firefox browser and accesses `http://localhost:3333/getActivePlayers`
3. The user shuts Firefox down.
4. The user opens Chrome and accesses `http://localhost:3333/getActivePlayers`.

How many different sessions does the Node.js server maintain at the end of these actions?

- A. 0
- B. 1
- C. 2
- D. 3

Note: while answer (C) was meant to be the correct answer, due to typo in line 46 (es instead of res), we give the point for both answers (A) and (C).

Question 38

WCQ Looking into WCQ's server-side script, you come across the following route handler:

```
1 app.get("/randomColors/:i", function (req, res) { /* ... */ });
```

What is the purpose of `:i` in the definition of the route?

- A. `:i` is a routing parameter; its value can be accessed via `req.params.i`.
- B. `:i` indicates that the route is case-insensitive, i.e. the paths `/RandomColors`, `/randomCOLORS`, `/RaNdOmCoLoRs` and any other combination of corresponding lower and upper-case letters match this route.
- C. `:i` refers to the URL query parameter name `i`. The query parameter value can be accessed via `req.query[i]`.
- D. No special meaning is attached to `:i`. The route handler matches just a single path: `/randomColors/:i`

Question 39

WCQ Consider a user who opens a browser tab T and accesses WCQ's insights screen by typing `http://insights.mydomain.nl:3000/` into the address bar. The following cookie is sent in the HTTP response:

```
Set-Cookie: insight-views=1; Domain=mydomain.nl; Path=/summary
```

Over the next minute, the user enters the following five URLs into tab T:

- `http://insights.mydomain.nl:3000/`
- `http://insights.mydomain.nl:3000/summary`
- `http://mydomain.nl:3000/`
- `http://mydomain.nl:3000/summary`
- `http://summary.mydomain.nl:3000/`

How many times is the cookie defined above sent in an HTTP request?

- A. 1
- B. 2

C. 3

D. 4

Question 40

GENERAL Assume that the source code below is stored in a file and opened in a modern browser. Which of the following statements is true?

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <style>
5     body {
6       background-color: gold;
7       height: 100vh;
8       width: 100vw;
9     }
10  </style>
11  <title>Color test</title>
12 </head>
13
14 <body id="b">
15   <script>
16     const colors = ["violet", "tomato", "crimson", "darkred", "seagreen"];
17     function setColor() {
18       const random = Math.floor(Math.random() * colors.length);
19
20       if (document.cookie.includes("cook=") == false) {
21         document.body.style.backgroundColor = colors[random];
22         document.cookie = "cook=" + colors[random] + ";max-age=5;httpOnly";
23       }
24     }
25     document.getElementById("b").addEventListener("click", setColor);
26   </script>
27 </body>
28 </html>
```

- A. A click on the browser viewport that happens five or more seconds after the last background color change, will lead to a change of background color. Other clicks have no effect.
- B. Clicks on the browser viewport will change the background color, no matter how many seconds pass between clicks.**
- C. The background color will not change in response to any clicks as the `addEventListener` method has no event type `click`.
- D. After the first click on the browser viewport, the background color will change once and remain in this color, independent of the number of clicks that follow and when those clicks occur.

END OF EXAM.