IWP: Internetwork- and web-programming

Written exam, June 8, 2020. Aalborg University

- You must upload you answers in a pdf-file to the "Digital-exam" system.
- Write your name and study-number at the front page
- You need not repeat the exercise text. It is sufficient to clearly identify the question using the question numbers in this assignment-sheet.
- You can produce the required pdf file using a normal word or text processing system and then using either its save-as-pdf or print-to-pdf, depending on the features of your chosen word processor. Alternatively, use a pdf-annotator to directly give your answers in the sheet (do not use yellow sticker notes that must be explicitly "clicked" to open to reveal their text).
- It is recommended that you read through the assignment sheet at the beginning of the examination to prioritize your time with respect to the number of points. While answering, ***monitor your time*** to ensure that you do not get sidetracked or stuck (especially in the practical part).
- If you think there is a mistake in an assignment, or lacks information, please state your assumptions with the answer.

This exam set contains 6 main assignments, each with a number of sub-questions. You can collect a total of 100 points.

1. HTML and HTTP (15 pts) 2
2. Computer Networks and the application layer (10 pts) 5
3. Reliable Data Transfer and the Transport Layer (16 pts) 7
4. The Network and Link Layer (16 pts) 10
5. Network and web-security (10 pts) 13
6. Web-programming - Practical Assignment (33 pts) 14

Michele Albano and Brian Nielsen

1. HTML and HTTP (15 pts)

Question 1.1

Q1.1.1 What is the correct HTML for creating a hyperlink?

```
1. <a href="http://www.cs.aau.dk">Computer Science</a>
2. <a url="www.cs.aau.dk "> Computer Science </a>
3. <a name="http://www.cs.aau.dk">www.cs.aau.dk</a>
4. <a>http:// http://www.cs.aau.dk </a>
```

Answer	1

Q1.1.2 What is the correct HTML for making a text input field?

```
1. <textinput type="text">
2. <input type="text">
3. <input type="textfield">
4. <textfield>
```

Answer	2

Q1.1.3 Which doctype is correct for HTML5?

```
1. <!DOCTYPE html>
2. <!DOCTYPE HTML5>
3. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0//EN"
   "http://www.w3.org/TR/html5/strict.dtd">
```

Answer	1

Q1.1.4 Where is the correct place to insert a JavaScript?

1. In the `<head>` section
2. In the <body> section
3. Both are correct

Answer	3

Q1.1.5 Inside which HTML element do we put the JavaScript?

```
1. <script>
2. <javascript>
3. <js>
4. <scripting>
```

Answer	1

Q1.1.6 What is the correct JavaScript syntax to change the content of the HTML element below?

```
<p id="demo">This is a paragraph of text.</p>
```

1. `1document.getElementByName("p").innerHTML = "Hello World!";`
2. `#demo.innerHTML = "Hello World`
3. `document.getElementById("demo").innerHTML = "Hello World”;`
4. `document.getElement("p").innerHTML = "Hello World`

Answer	3

Question 1.2

Consider the following sequence of request and response messages between a client and a web-server:

```
GET / HTTP/1.1\r\n
Host: test.com\r\n
Connection: keep-alive\r\n
Cache-Control: no-cache\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) \r\n
Accept: text/html, application/xhtml+xml, application/xml; \r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,da;q=0.8,nb;q=0.7,de;q=0.6\r\n
\r\n
```

```
HTTP/1.1 200 OK \r\n
Connection: keep-alive\r\n
Content-Type: text/html;charset=utf-8\r\n
Etag: "15f0fff99ed5aae4edffdd6496d7131f"\r\n
Content-Length: 379\r\n
Date: Tue, 26 May 2020 11:06:55 GMT\r\n
\r\n
```

```
GET / HTTP/1.1\r\n
Host: test.com\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Accept: text/html, application/xhtml+xml, application/xml; \r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,da;q=0.8,nb;q=0.7,de;q=0.6\r\n
If-None-Match: "15f0fff99ed5aae4edffdd6496d7131f"\r\n
\r\n
```

```
HTTP/1.1 304 Not Modified \r\n
Content-Length: 0\r\n
Connection: keep-alive\r\n
Etag: "15f0fff99ed5aae4edffdd6496d7131f"\r\n
Date: Tue, 26 May 2020 11:07:00 GMT\r\n
\r\n
```

Q1.2.1

Question	Answer
1. Which http method is used?	GET
2. Which protocol is used?	http v 1.1
3. What kind of document does the server send in its reply	Html document

Q1.2.2

Why does the server respond with code 304 to the second request?

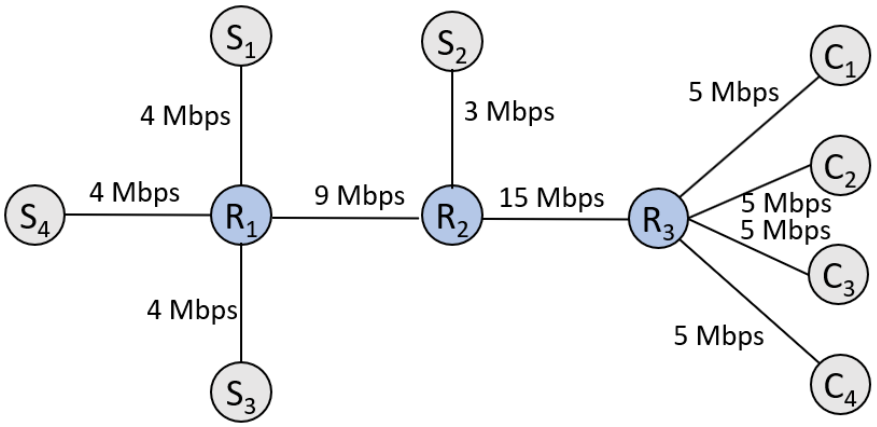
1. `Content-type` does not match the request.
2. `Cache-control` set max-age in the second request to 0
3. The request is unnecessary as the connection is kept alive (`Connection: keep alive`)
4. The client already received the requested document 5 seconds ago (`Date`: xxx).
5. The server's version of the document matches that of the client.
6. The document was deleted at the server

Answer:	5, because of If-None-Match header and matching E-tags

2. Computer Networks and the application layer (10 pts)

Question 2.1

Consider the network shown in the figure below that shows 4 servers (S_1… S_4) and 4 clients (C_1… C_4). The network is interconnected using the routers (R_1… R_3) with the possible transmission rates for each link. The clients simultaneously stream video from the corresponding server such that C_1 streams from S_1, C_2 streams from S_2 etc.



What is the possible end-to-end throughput for each client?

Answer	3 Mbps, R1-R2 link is the bottleneck; it is reasonable to assume that the bps is shared fairly.

Question 2.2

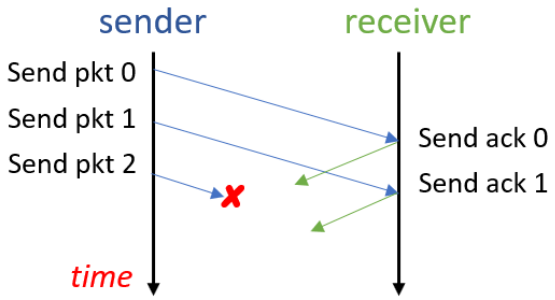
Consider the Internet protocol stack. Which layers does the following tasks belong to? Mark by A (Application Layer), T (Transport Layer), N (Network Layer), L (Link Layer), P(Physical). If the task is handled by several layers, assign the letters for all layers involved.

Task	Layer (A, T, N, L, P)
Translating a domain name into an IP address	A (DNS is an application layer service)
Translating an Ethernet-address to an IP address	L, (N) (ARP, Reverse ARP are at Link layer, DHCP is presented as part of Network layer)
Performing congestion control	T, (N) (T via TCP, N via ICMP)
Performing flow control	T, (A), (L) Primarily TCP, but for some link layers
Performing reliable delivery	T, (L) Primarily TCP, but also some lossy link layer protocols, eg. for satellite links
Computing link cost	N (Link cost is use by routing protocols)
Doing 3-way handshakes	T (TCP connection establishment)
Forwarding packets	N, L (Both!, Forwarding IP datagrams, Ethernet frames, see Fig 1.24)
Providing End-to-End encrypted communication	A (Encryption is not supported by the Internet protocol stacks transport layer) TLS/SSL are application layer.

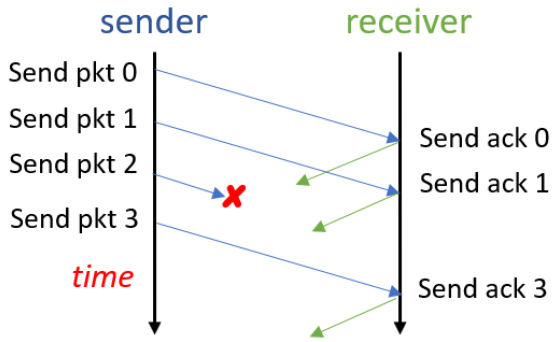
Additionally flow/congestion control, reliable delivery forwarding (as in P2p networks) may be done as part of the application, but not primarily.

3. Reliable Data Transfer and the Transport Layer (16 pts)

Question 3.1



Protocol (a)



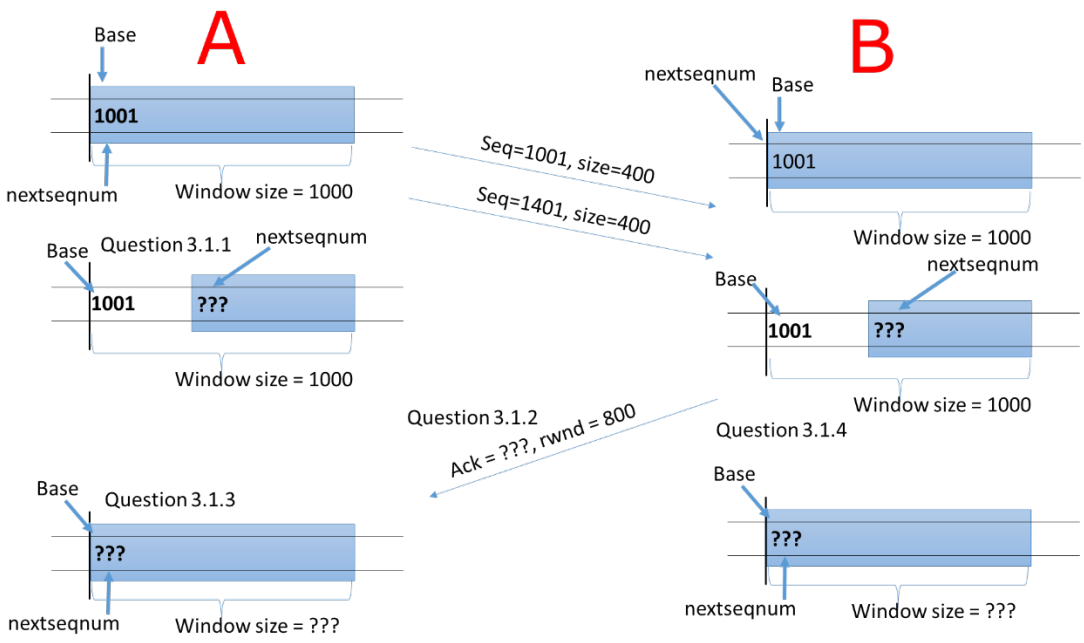
Protocol (b)

The above figure shows two sliding window protocols (a) and (b). For each protocol, indicate whether Go-Back-N, Selective Repeat is being used, or if insufficient information is available. Set "X".

	Protocol (a)	(Protocol (b)
Go-Back-N		
Selective Repeat		X, as receiver acks pck 3
Insufficient Information	X	

Question 3.2

Consider the following exchange of segments between A and B, where no message is lost. As usual, the TCP window of the sender is defined by the *base, nextseqnum* and the *window size* (see for example figure 3.19 of page 250 of the Kurose&Ross textbook). A packet containing data reports a sequence number with byte granularity and the size of the data. An ACK packet specifies what is being ACKed and it can set the new size of the TCP window.



Q 3.2.1: After A sent two segments of 400 bytes each, what is the "nextseqnum" of A's window?

1. 1400
2. 1401
3. 1800
4. 1801

Answer	4 (1801), since "nextseqnum" points to the next byte that will be sent, and the two packets have sent bytes 1001 to 1800.

Q 3.2.2: What is the value of the ACK field of the ACK packet?

1. 1001
2. 1401
3. 1801
4. 2001

Answer	3 (1801) since "Ack" points to the next byte that B wants to receive, and from the figure we see that he received correctly the two previous datagrams from A.

Q 3.2.3: Please complete the window of A with (base, nextseqnum, window size):

1. 1001, 1001, 800
2. 1401, 1401, 800
3. 1801, 1801, 800
4. 1801, 1801, 1000

Answer	3 (1801, 1801, 800). Everything up to byte 1800 was ACKed (thus Base = 1801), the next byte we have to send is 1801, and the size of the window is now 800, as requested by the ACK datagram from B.

Q 3.2.4: Is there any difference between the second windows of A and B, or between the third windows of A and B?

1. No
2. In the second window, before sending the ACK packet B has still a 1401. Thus, the second window would have base, nextseqnum and window size = 1001, 1401, 1000
3. In the third window, B has still a large buffer, thus base, nextseqnum and window size = 1801, 1801, 1000

Answer	1 (No). There were no lost packets, thus the views of A and B are aligned.

Exercise 3.3 What information defines TCP socket and a UDP socket?

1. IP address and port of the sender. UDP: IP address and port of the receiver.
2. TCP: IP address and port of both sender and receiver. UDP: IP address and port of the sender.
3. TCP: either IP address and port of the sender, or IP address and port of both sender and receiver. UDP: IP address and port of the sender.

Answer	3: the UDP socket is defined only by the sender's IP address and port, while for TCP the socket can be a "server socket"/"welcome socket", thus with just local IP address and local port, or it can be a "bound socket", with both local and remote IP addresses and ports. So 3 is most accurate, but we accept 2 as well, given that – part of multiplexing (connected sockets)—the book states it is the 4 tuple <sender Ip+port, receiver IP+port)>

4. The Network and Link Layer (16 pts)

Question 4.1

Consider that A is sending packets to B, and B sends back ACKs. Consider that the network is lossy and 10% of the packets from A to B are lost, and that the loss on ACKs is negligible. How many packets are sent by A on average, for each packet correctly received by B?

1. 0.1
2. 1
3. 1.1
4. Between 1.11 and 1.12

Answer	4: in 90% of the cases, 1 packet. In 90% of the other 10%, 2 packets. Anyway, in 1% of the cases you need at least 3 packets, since both the original packet and the first retransmission were lost. Depending on how many retransmissions are allowed for a lost datagram, between 1.111 and 1.11111…. are needed on average. We also give partial points for 3.

Question 4.2

Consider TCP communication, and datagrams having headers of 20 bytes for the TCP protocol and 20 bytes for the IP protocol. An IP datagram of 2300 bytes is sent over wifi (MTU 2304), and it gets to an Ethernet network (MTU 1500). The datagrams gets fragmented. How much overhead there is in the two networks, with respect to the application layer?

1. 20 bytes in the first network, 40 bytes in the second network.
2. 20 bytes in the first network, 30 bytes in the second network.
3. 40 bytes in the first network, 60 bytes in the second network.
4. 40 bytes in the first network, 80 bytes in the second network.

Answer	3, since in the first network there is one packet of application data with TCP header (20 bytes) and IP header (20 bytes). In the second network there are two packets, both of them with IP header, but the TCP header is not repeated since the fragmentation was done in the network layer.

Question 4.3

A router has the following routing table:

Network Destination	Netmask	Interface	Gateway
192.168.10.0	255.255.255.0	eth0	----------
192.168.11.0	255.255.255.0	eth1	----------
0.0.0.0	0.0.0.0	eth2	192.168.15.1

Q 4.3.1: What happens to a packet with destination address 8.8.8.8 received on interface eth0?

1. it is dropped
2. it is delivered through interface eth0
3. it is sent to 192.168.15.1 via interface eth2

Answer	3, since the destination address does not match any other rule except for the default one.

Q4.3.2: What happens to a packet with destination address 192.168.10.1 received on interface eth2?

1. it is dropped
2. it is delivered through interface eth0
3. it is sent to 192.168.15.1 via interface eth2

Answer	2, since the destination address matches the first rule.

Question 4.4

A company has bought the IP domain 199.59.242.0/26, and it needs to subnet it to its 3 departments, which need respectively 20, 16 and 3 IP addresses. Which sentence is correct, regarding the subnetting strategy?

1. The first group can get 199.59.242.0/28, second can get 199.59.242.32/29, and the third gets 199.59.242.48/30.
2. The first group can get 199.59.242.0/27, second group can get 199.59.242.32/28, the third can get 199.59.242.48/29.
3. The first group needs a /27 subnetwork, the second group needs a /27 subnetwork, the third group needs a /29. The /26 domain bought by company is not enough, and there is no solution to the subnetting problem.

Answer	3, and in fact note that, if the host part of the address is x bytes, you can accommodate up to $2^x - 2$ hosts in there. Thus, for 20 and 16 hosts you need /27 networks. (If you request e.g. 3 IP addresses from your sysadmin, you expect that you get e.g. 3 IP addresses that can be used for hosts.)

5. Network and web-security (10 pts)

Question 5.1

Imagine a software has a defect, it sends out a private key, but keeps the corresponding public key secret. Which sentence is *false*?

1. Everything can work, if the private key is used in all the certificates, etc., and the public key is kept secret at all times.
2. Security can be weaker, since a public key tends to be shorter, thus it is easier to find when the private key is known.
3. It is not possible to put anything different from an original public key in a certificate.

Answer	3, since private and public keys can be switched, but this can lead to weaker security.

Question 5.2

Compare ApiKeys and OAuth2. Which of the following is an advantage of ApiKeys over OAuth2?

1. 1 – They allow to authorize a client for a short period of time.
2. 2 – Their usage is easier to program in a software.
3. 3 – They can provide a clear distinction between authentication, and authorization.

Answer	2, since 1 and 3 are among the advantages of OAuth2 over ApiKeys.

6. Web-programming - Practical Assignment (33 pts)

A group of friends wish to ensure that they do not get drunk at parties. They therefore request a web-app to help them monitor their blood alcohol content (BAC). The app should have two main functionalities

1. Allow a user (identified by name) to report having consumed a drink.
2. Allow the user to track the BAC of (potentially another) person in the group.

BAC is the amount of alcohol per volume blood, typically measured in per-mille (DK: "promille"). A simple model[1] for estimating the BAC can be computed according to the formula below (assuming an initial BAC of zero).

$$BAC = \frac{alcohol\ (grams)}{weight\ (kg) \times F} - 0.15 \times T$$

- *Alcohol* is the amount of alcohol introduced by the drink. In this assignment a drink (DK "en genstand") equals 12 grams.
- *F* is a constant that represents physiological differences between men and female: it is 0.6 for female, and 0.7 for male.
- *T* is the amount of time elapsed since the drink was consumed.

For the assignment we assume that BAC calculation is additive, i.e., if a drink was consumed at t_1 and another drink is consumed at time t_2 (t_2 >= t_1), then the BAC evolves as the sum the BAC from each individual drink. Remark, that BAC never becomes negative.

The file (download appendix from digital exam) `NodeWeb-BAC.zip` contains a skeleton for this application. Unzip the archive to a suitable location on your machine. The goal of the assignment is to add certain functionality to the skeleton.

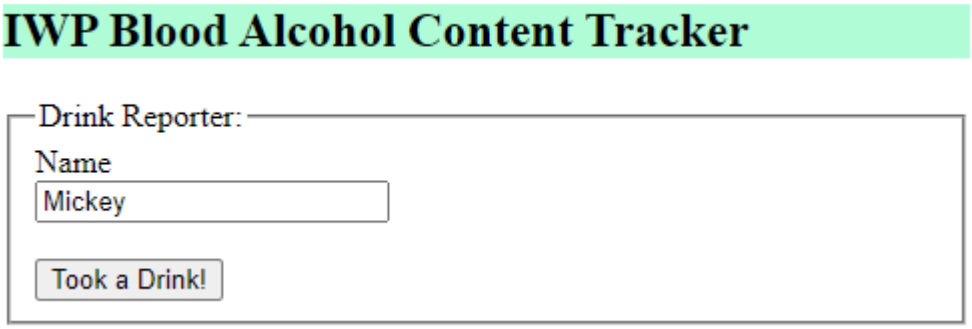
A small node.js application (bac-`app.js`) acts as a http server that serves the front-page (stored in `bac.html`) and a web-API function that allows the app to store and retrieve JSON objects containing the information for BAC calculation. The information is stored in an in-memory "database" implemented as a simple array of records (`bacDB`). A record is a JavaScript object with the following properties *name, gender, weight, drinkTime;* drinkTime is a `Date` object containing the time that the server recorded the drink consumption. In the current state, the server accepts an HTTP POST to the URI `/bac-records` that stores a new record with information that the user has entered.

When the server is started (either run the app.js from within VisualStudioCode, or the command line

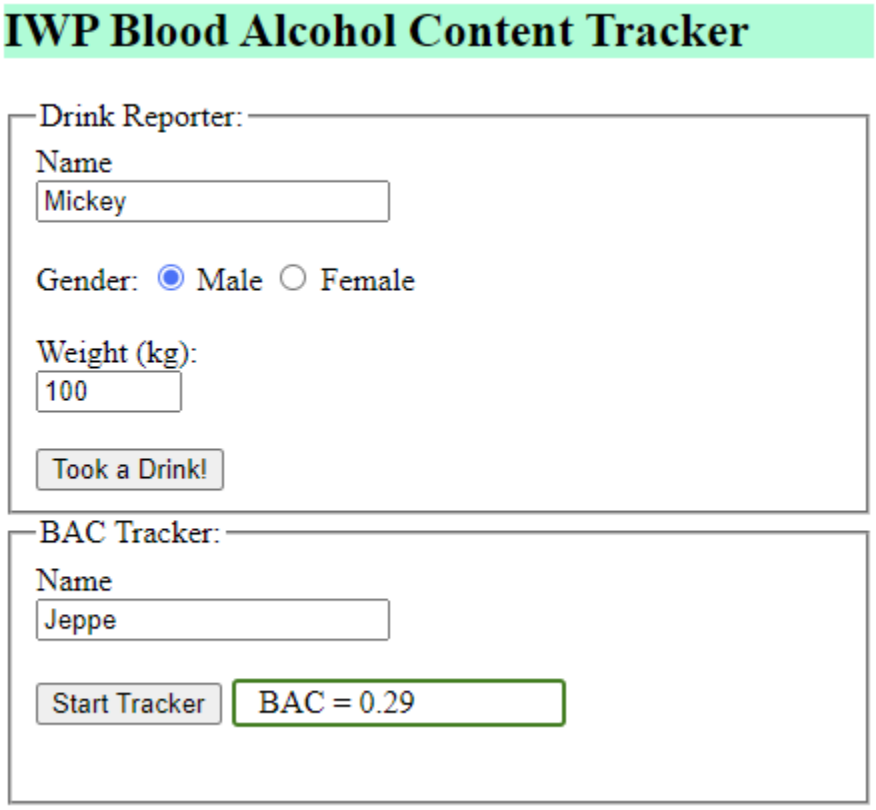
NodeWeb-BAC >`node node/bac-app.js`

and a web browser is pointed to the localhost (127.0.0.1) at port 3000, the browser should show the following page:

[1] Source: https://www.sundhed.dk/borger/patienthaandbogen/psyke/sygdomme/alkohol/alkoholpromille-beregning/



At the end of the assignment your frontpage should appear somewhat like the screenshot below: In the first part reports to the app that the named person has consumed a single drink at the time of submission. The second part *periodically* updates the named person's BAC from the server.



Remark, that the questions can be answered partially independently! If you get stuck in one step, try to proceed using fixed/dummy/stubs data and functions.

Question 6.1:

Extend the form with fields to enter weight (kg) fields and gender. Use *HTML validation* to ensure that they are filled out and satisfy the constraints: 1<=weight<=300, and gender is mandatory. Add the form for the tracker part. Do not spend time on layout and styling, but you are welcome to use the style in the attached stylesheet.

```
  <form id="bacDrinkForm" action="bac-records" method="post">
    <fieldset>
      <legend>Drink Reporter:</legend>
      <label for="name"> Name</label>
      <input type="text" name="name" id="name_id" placeholder="Mickey" required minlength="1" max
length="30">
      <label for="gender" > Gender:</label>
      <input type="radio" id="male_id" name="gender" value="Male" required>
      <label for="male">Male</label>
      <input type="radio" id="female_id" name="gender" value="Female">
      <label for="female">Female</label>

      <label for="weight"> Weight (kg):</label>
      <input type="number" name="weight" id="weight_id" placeholder="70" min="1" max="300" requir
ed>
  <input type="submit" id="drinkBtn_id" value="Took a Drink!">
</fieldset>
</form>
<form id="bacTrackForm" action="bac-records" method="get">
  <fieldset>
    <legend>BAC Tracker:</legend>
    <label for="trackerName"> Name</label>
    <input type="text" name="trackerName" id="trackerName_id" placeholder="Jeppe" required minlen
gth="1" maxlength="30">
    <input type="submit" id="trackBtn_id" value="Start Tracker"> <output id="bacResult_id"> BAC =
 ??? </output>
   <output id="" style="visibility:hidden">Error: </output>
</fieldset>
</form>
```

Input type for weight should be = number, with required min,max (else it must be handled explicitly in JS)
Required gender field
Name fields must have unique id's!

It is OK to use select instead of radio input.It is OK to add prefilled "values" or "placeholders".

Check existence of TrackerForm, and field to store output.
(Method of trackform or in JS; name attributes in name, weight not strictly required as it is not shipped as form-submit; As the assignment does not ask for CSS, using
 to make forced linebreaks to layout form is accepted)

Question 6.2

When the form is submitted, a JavaScript function at the client is to extracts the drink data information in the form-fields (currently only name) and returns a JavaScript object. Augment the function to also extract weight and gender. Show the augmented function below:

Hint: If you use radio buttons, getting their state information can be done in numerous ways. A crude but fully acceptable solution is to assign an id to each input choice and then test its "checked" property.

```
function extractDrinkData(){
  let bacDrinkData={};
  bacDrinkData.name=document.getElementById("name_id").value;
  let maleChoiceElement=document.getElementById("male_id");
  let femaleChoiceElement=document.getElementById("female_id");
  if(maleChoiceElement.checked){
    bacDrinkData.gender=maleChoiceElement.value;
  };
  if(femaleChoiceElement.checked){
    bacDrinkData.gender=femaleChoiceElement.value;
  };

  bacDrinkData.weight=Number(document.getElementById("weight_id").value);
  console.log("Extracted"); console.log(bacDrinkData);
  return bacDrinkData;
}
```

Check that correct HTML elements are identified.

Check that values are inserted into returned object.

Check that the checked attribute is used to identify gender (or that another correct method is used.) Simply reading value is insufficient.

Question 6.3

At the server bac-app.js a POST to the resource `/bac-records` results in that the received JSON object is converted into a JavaScript object with the new submitted drink data, validated, and then conditionally inserted in the "data base". Show the updated validation function that validates the new weight and gender fields.

```
function validateBacDrinkForm(bacDrinkFormData){
  let nameLen; let name; let weight; let gender;
  try {
    nameLen=bacDrinkFormData.name.length;
    name=bacDrinkFormData.name;
    weight=Number(bacDrinkFormData.weight);
    gender=bacDrinkFormData.gender;
  }
  catch(e) {console.log (e);throw (new Error(ValidationError));}

    if((nameLen>=minNameLength) && (nameLen<=maxNameLength) &&
       (minWeight <= weight) && (weight <=maxWeight) &&
       ((gender===MaleGender)||(gender===FemaleGender))){
      let drinkData={name: name, weight:weight, gender: gender};
      return drinkData;
    }
    else throw(new Error(ValidationError));
 }
```

Check weight bounds and gender

Check that drinkData includes 3 "validated" attributes name,weight, gender

Remark that server-side validation is absolutely mandatory since there is no guarantee that data origins from the client that you have programmed and supplied; Another client application (or simply postman) can send harmfull/insecure data.

Question 6.4

Implement a Javascript function `calcBAC(name)` that computes the current BAC for the person named "name". If you wish to skip this question for now, simply return a fixed value, e.g. 2.

Hint: The skeleton contains a helper function that computes the time difference in milli-seconds between two `Date` timestamps.

```
function calcBAC(name){
  console.log("computing "+name);
  let bacSum=0;
  let now=new Date();
  for(let i=0;i<bacDB.length; i++){
    if(bacDB[i].name===name) {
      let drinkTime=bacDB[i].drinkTime;
      let duration=dateDiffInHours(now,drinkTime);
      let bacIncrement=drink2BAC(bacDB[i].weight,bacDB[i].gender);
      bacSum+=burnDrink(bacIncrement,duration);
    }
  }
  return round2Decimals(bacSum);
}

function burnDrink(initialBAC,duration){
  return Math.max(0,initialBAC-0.15*duration);
}
const ALCOHOL_IN_DRINK=12;
function drink2BAC(weight,gender){
 if(gender===MaleGender)
     return ALCOHOL_IN_DRINK/(weight*0.7)
 else
   return ALCOHOL_IN_DRINK/(weight*0.6);
}
```

Check that bac (BAC-0.15*duration) does not become negative.

Check gender difference

Check to remember duration since last drink to "now"
Check consumption accumulates with multiple drinks.

(This solution is over-simplified, but fully acceptable as per simplifying assumption in the assignment. The correct calculation needs to "simulate" the BAC level by consuming one drink at a time and "burning" the accumulated level (you can drink 3 beers at "once", but only burn them sequentially). See code below.)

```
function calcBAC(name){
  console.log("computing "+name);
  let i=0;
  let bac=0;
  let previousDrinkTime;
  let foundFirst=false;
  //find first drink
  while(i<bacDB.length && !foundFirst){
   if(bacDB[i].name===name) {
      foundFirst=true;
      previousDrinkTime=bacDB[i].drinkTime;
      bac=drink2BAC(bacDB[i].weight,bacDB[i].gender);
    }
    i++;
  }
  if(!foundFirst) return null;//name not found.
  //add effect of subsequent drinks
  for(;i<bacDB.length; i++){
    if(bacDB[i].name===name) {
      let drinkTime=bacDB[i].drinkTime;
      let duration=dateDiffInHours(drinkTime,previousDrinkTime);
      bac=burnDrink(bac,duration);
      bac+=drink2BAC(bacDB[i].weight,bacDB[i].gender);
      previousDrinkTime=drinkTime;
    }
  }
  //and burn BAC untill "now"
  let duration=dateDiffInHours(new Date(),previousDrinkTime);
  bac=burnDrink(bac,duration);
  console.log("CALC "+bac);
  return round2Decimals(bac);
}
```

Question 6.5

Extend (the server's web-api) such that a HTTP call to the endpoint `/bac-records/name` returns a JSON object to the client containing the current BAC for the person named "name", e.g., a call to `/bac-records/Mickey` should return BAC for "Mickey" at the time of the request.

Q6.5.1 Which HTTP method would you use

1. GET, since this is a method that does not alter the DB
2. POST because it involves an expensive computation at the server, and need to be recomputed for each request (as more time has elapsed)
3. GET, but disables the caching of the response, and I consider the computation light.
4. Implement my own dedicated HTTP method.

Answer:	2 or 3. 3 is most correct (it is not modifying the db, but still does a computation). The bac computation incurs (using a good db and calculation alg.) a light server load. But, as the result depends on the real-time clock, care must be taken to prevent clients from caching it. Answer 2 is also acceptable (as it is a non-idempotent operation), or a poor bac alg with a large db could incur high load. Note: not modifying a db is an insufficient condition to use GET.

Q6.5.2 Show here your code for the route-handling and possible extra functions that you have added.

```
…
    case "GET":{
…
            case "bac-records":
```

```
        if(pathElements.length===3 && lookup(pathElements[2])){ // handle url of t
he format "/bac-records/name"
          res.setHeader('Cache-Control', 'no-store');
          return jsonResponse(res,{name: pathElements[2], bac: calcBAC(pathElement
s[2])});
        }
        else return errorResponse(res,404,"No Such Resource");
        break;
```

Primarily check that the right place of change (route) is identified.

Check that a json response is returned, including a computed BAC.

Should be consistent with 5.2.1 (if POST is used then the code should be in a different position from the template. Should be visible in the answer).

Check that a 404 error is (only) sent with the resource does not exist.

Setting cache-header is not mandated by the solution (but should be for a production app).

Question 6.6

Q6.6.1 Implement the client-side functionality for the tracker function. i.e. fetch the current BAC and update the HTML element that is to contain the result. Additionally, an entry to the console log containing the resulting BAC should be added when a new response is received.

HINT: A helper function to fetch json objects is provided. First perform a single fetch, and add the result to the log. Then update the element to contain information like "*BAC = <bac>"*. As a third step, set up an interval timer to periodically update the element, that starts running when the tracker is started. You need not implement functionality for stopping the tracker in this version.

Even if your fetch is not working correctly, the later steps can be completed using a fixed/dummy data at the client side.

```
let trackerTimer=null;
function updateBAC(name){
  jsonFetch("bac-records/"+name).then(bacResult=> {
    console.log(bacResult);
    document.getElementById("bacResult_id").textContent="BAC = "+bacResult.bac;
  } ).catch(e=>{
      console.log("Ooops "+e.message);
      alert("Error getting "+name);
      clearInterval(trackerTimer);})
}
```

```
function startTracker(event){
  event.preventDefault(); //we handle the interaction with the server rather than
browsers form submission
  let name=document.getElementById("trackerName_id").value;
   updateBAC(name);
   if(trackerTimer!=null) clearInterval(trackerTimer);
   trackerTimer=setInterval(() => {
     updateBAC(name);
   }, 1000);

}
document.getElementById("bacDrinkForm").addEventListener("submit", sendDrink);
document.getElementById("bacTrackForm").addEventListener("submit", startTracker);
```

Check that bac-records+name is fetched

Check that the (chosen) result element is updated

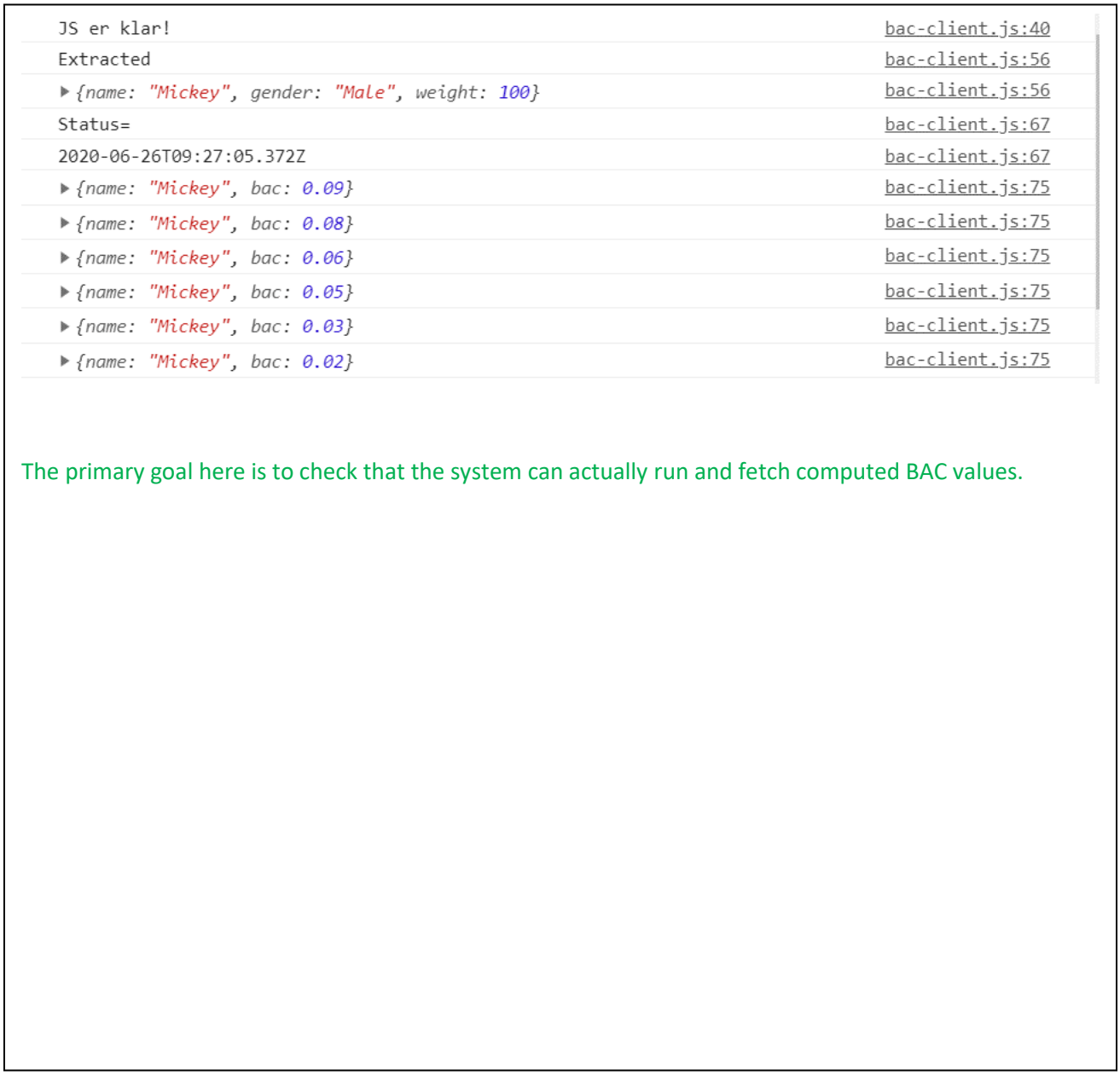
Check that an appropriate event listener for the submit button of the tracker is added.

Check addition of interval timer. It is OK if the first fetch doesn't start until the first timer fires.

Clearing time is not mandated by the assignment.

Question 6.6.2

Show the client-side log contents resulting from the above code: (you may insert text or an image).



The primary goal here is to check that the system can actually run and fetch computed BAC values.