IWP: Internetwork- and web-programming

Written exam, August 25, 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.
- The distribution of points to assignments is based on our judgement from a combination of difficulty, expected time needed, and importance.
- *If you think there is a mistake in an assignment, or lacks information, please state your assumptions with the answer.*

This exam set contains 5 main assignments, each with a number of sub-questions. You can collect a total of 100 points.

1. HTML and HTTP (24 pts) 2
2. Computer Networks and the Application layer (17 pts) 7
3. Reliable Data Transfer and the Transport Layer (18 pts) 10
4. Network Security (8 pts) 15
5. Practical Assignment (33 pts) 16

Michele Albano and Brian Nielsen

1. HTML and HTTP (24 pts)

Question 1.1 (7,5 pts)

Q1.1.1 Which tag is used to mark the beginning and end of an html document?

1. DOCUMENT
2. HEAD
3. HTML
4. BEGIN

Answer	

Q1.1.2 Which purpose does the attribute “name” serve on an HTML input element? State all correct answers.

1. Generates a name/value pair consisting of the given name and the value of the element
2. Allows a stylesheet to identify the named elements
3. Defines a text-label with the specified name for the input field
4. Name must be present to include the input fields value in a submitted form

Answer	

Q1.1.3 The purpose of HTML5 is to

1. Define styling and layout of a web-document
2. Permit a client and server to exchange information on the Internet
3. Define semantics of content and structure of a hypertext document
4. Markup prices in a pricelist

Answer	

Q1.1.4 What is the correct HTML syntax for creating a hyperlink with the text “good science reading” to the destination “sciencemag.org“, to be accessed using the secured hypertext transport protocol.

Answer	

Q1.1.5 What output does the following JavaScript program fragment output to the console?

```
console.log("Start");
fetch("/date").then(date=>console.log("Got Reply"));
console.log("Finished");
```

Answer	

Question 1.2 (5 pts)

Consider the following headers from a request and resulting response messages between a client and a web-server:

```
POST /api/getmeasure HTTP/1.1\r\n
Host: app.netatmo.net\r\n
Connection: keep-alive\r\n
Content-Length: 184\r\n
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google
Chrome";v="90"\r\n
Accept: application/json, text/javascript, */*; q=0.01\r\n
Authorization: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\r\n
sec-ch-ua-mobile: ?0\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36\r\n
Content-Type: application/x-www-form-urlencoded; charset=UTF-8\r\n
Origin: https://my.netatmo.com\r\n
Sec-Fetch-Site: cross-site\r\n
Sec-Fetch-Mode: cors\r\n
Sec-Fetch-Dest: empty\r\n
Referer: https://my.netatmo.com/\r\n
Accept-Encoding: gzip, deflate, br\r\n
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,da;q=0.7,nb;q=0.6,de;q=0.5,en-GB-
oxendict;q=0.4,en-AU;q=0.3,en-CA;q=0.2,en-NZ;q=0.1,en-ZA;q=0.1\r\n
\r\n
scale=max&device_id=70%3Aee%3A50%3A2b%3A59%3Aca&module_id=02%3A00%3A00%3A2b%3
A1f%3Aaa&type=Temperature%2CHumidity&date_begin=1622193141&ci_csrf_netatmo=3e
a0c559f79522d238782ef4bcf9764e
```

```
HTTP/1.1 200 OK\r\n
Server: nginx\r\n
Date: Fri, 28 May 2021 13:03:44 GMT\r\n
Content-Type: application/json; charset=utf-8\r\n
Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
Cache-Control: no-cache, must-revalidate\r\n
Expires: 0\r\n
X-XSS-Protection: 1; mode=block\r\n
Access-Control-Allow-Origin: *\r\n
Strict-Transport-Security: max-age=31536000; includeSubDomains\r\n
X-Powered-By: Netatmo\r\n
\r\n
{"body":[{"beg_time":1622192801,"step_time":308,"value":[[13.8,84],[13.9,83]]
},...  data-removed ...
,"status":"ok","time_exec":0.08666515350341797,"time_server":1622207024}
```

Question	Answer
1. Which http method is used?	
2. Which protocol is used?	
3. Which parameters, if any, are sent to the server (names only)?	
4. Is the client allowed to cache the response? Why/why not?	
5. What kind of document does the server send in its reply	

Question 1.3 (6 pts)

Q1.3.1 HTTP is a stateless protocol. What does that mean (tick all that applies)?

1) Databases are not compatible with HTTP protocols
2) Session state must be sent from client to server with every request
3) Cookie can save state regarding a HTTP session on the client (browser)
4) Cookies can be stored on the server, and still HTTP is a stateless protocol
5) Resources can be created using HTTP POST requests, and still HTTP is a stateless protocol

Answer	

Q1.3.2 As HTTP is a stateless protocol, in the course we discussed how it is possible to store session data on the client-side (on the browser). Which of the following are client-side (browser's) storage options?

1) A web server
2) Cookies
3) A REST resource
4) A localStorage object
5) A certification authority

Answer	

Q1.3.3 Websockets is a mechanism for the server to send data to the client (browser) whenever the server wants to. Which of the following items are true regarding Websockets?

1) The client must expect that the connection from server to client gets closed, and the client is in charge of reconnecting it
2) It is necessary to buy a certificate from a certification authority to use Websockets commercially
3) Communication can happen on a port different from usual HTTP/HTTPS traffic
4) Browsers need to install a plugin / extension to use Websockets

Answer	

Question 1.4 (3 Pts.)

Q1.4.1 Consider that p is a Promise, and javascript code is doing:

```
p.then(result => continueComputation(result)).then(console.log)
```

In case the code executes without failures, what information can you infer regarding `continueComputation`? (For example, which data type `continueComputation` returns?)

Answer	

Question 1.5 (2,5 pts)

Q1.5.1 Consider the REST principles, and mark all the items that apply:

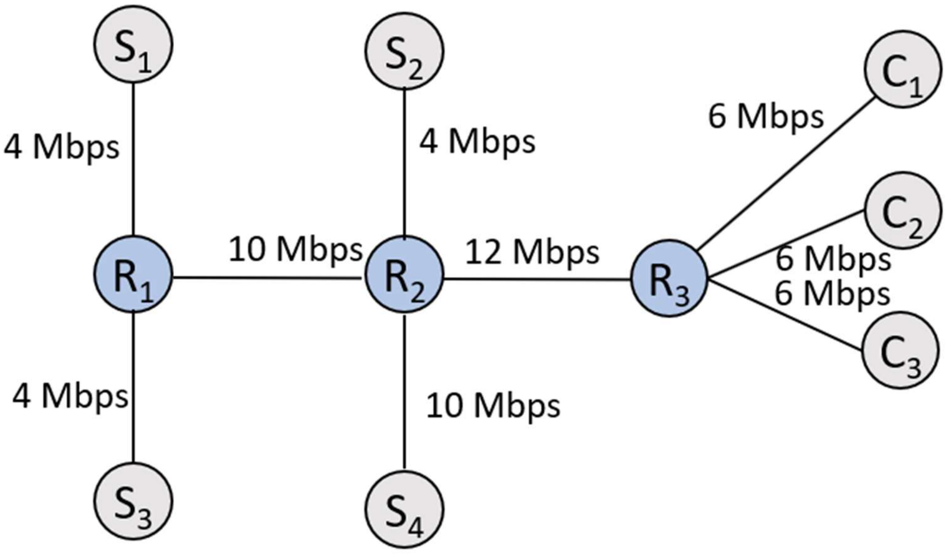
1. The “3001” in `https://127.0.0.1:3001/html/bovs.html` is the authentication code of the user
2. Even though REST is stateless, it is possible to store a new resource on the server after a POST request
3. Performing an idempotent request method one or two times has the same effect on the resource it is applied to
4. POST request methods must always be implemented as idempotent
5. A resource can be identified as `https://www.google.com/advanced_search`

Answer	

2. Computer Networks and the Application layer (17 pts)

Question 2.1 (4 pts)

Consider the network shown in the figure below that shows 3 servers (S_1… S_4) and 3 clients (C_1… C_3). 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	

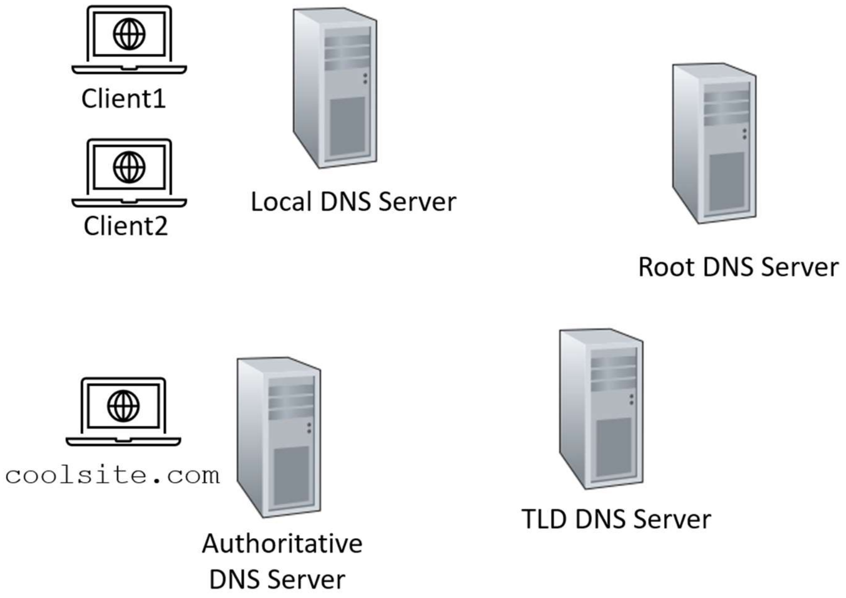
Question 2.2 (4 pts)

On what network device(s) are the application layer, transport layer, and network layer implemented? Mark with an "X".

Network device	Application Layer	Transport Layer	Network- Layer
Router			
Network interface Card			
Fiber Cable			
Switch			
End-system			

Question 2.3 (6 pts)

Consider the small DNS system illustrated below. Client 1 and client 2 runs on the same network and shares the same local DNS server. A new site `coolsite.com` is registered in the DNS system.



1. Client 1 wants access to `coolsite.com`. What sequence of request and replies are exchanged to perform the name lookup using an iterative query.
2. A little later, client 2 also wants access. What sequence of request and replies are exchanged to perform the name lookup using an iterative query.

Use the notation:

- A->B if host A sends a request to host B.
- A<-B if host B sends a reply to host A
- A<-> if host A sends a request to host B which immediately replies to A
- Sequence can be indicated by numbering or semi-colon: e.g: A->B; B->C

Answer 1	
Answer 2	

Question 2.4 (3 pts)

Socket communication. Think about the difference between TCP sockets and UDP sockets. Which of the following items apply to TCP sockets, but NOT to UDP sockets?

1. One TCP socket can be connected at the same time to 3 different computers
2. TCP communication initialization is fast, since there is no need for handshake between client and server
3. Some TCP sockets can be identified by (local IP address, local port, remote IP address, remote port)
4. The bytes received using TCP communication can be reordered with respect to how they were sent
5. There are cases when TCP communication can fail

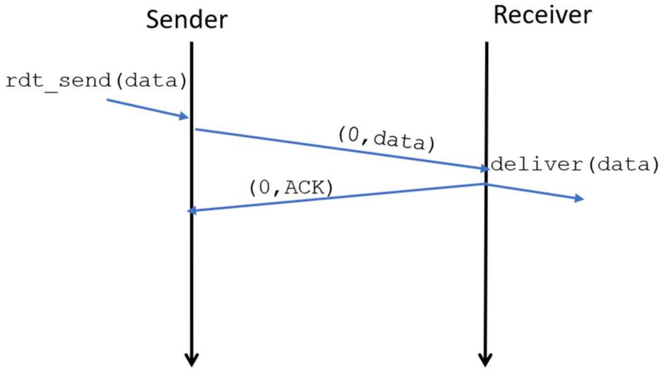
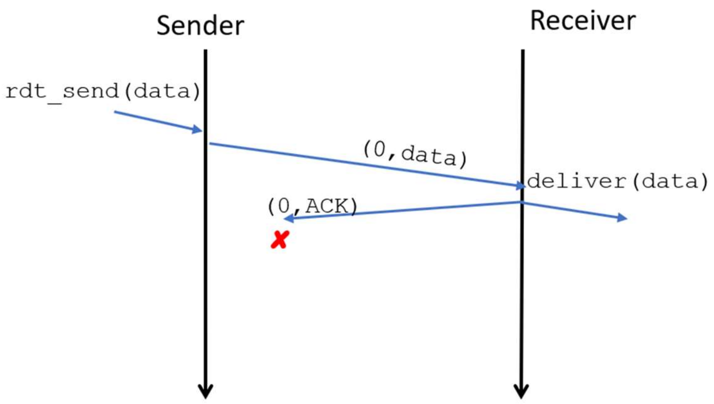
Answer	

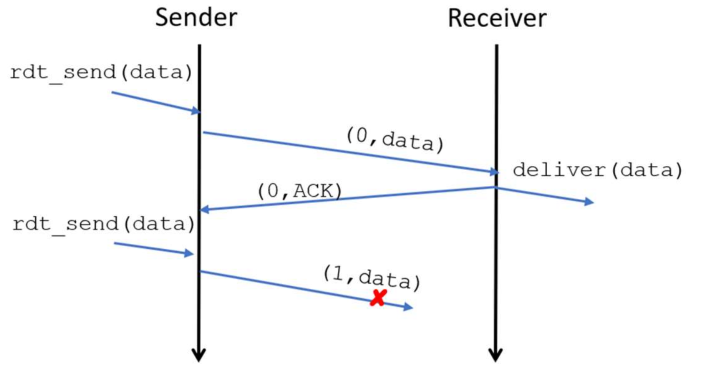
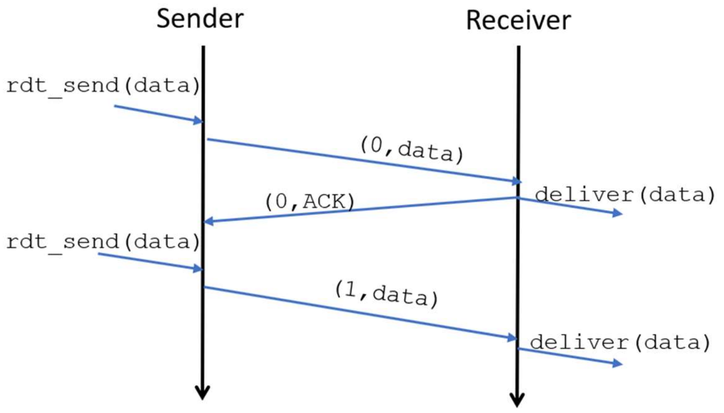
3. Reliable Data Transfer and the Transport Layer (18 pts)

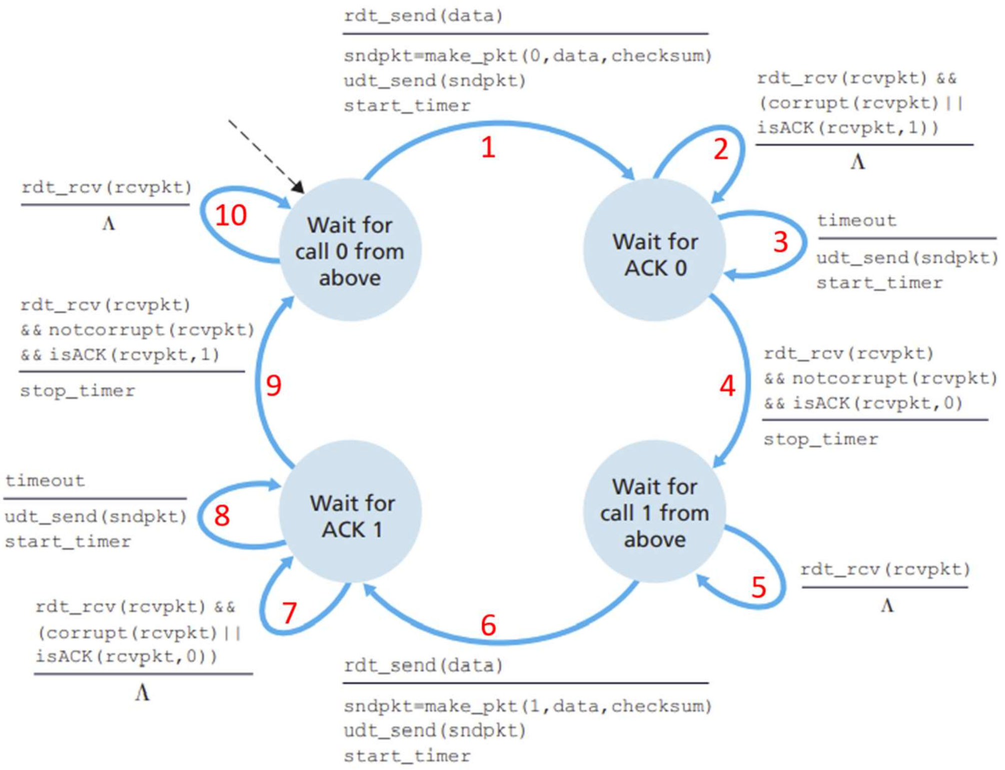
Question 3.1 (6 pts)

Two hosts use the alternating bit protocol (rdt3.0) to reliably transfer data. Use the state naming and transition numbering indicated in the below figure (adopted from Kurose&Ross Fig 3.15) to indicate for each of the below scenarios:

1. what is the name of the state of the sender after the last message it has sent or received (i.e. close to the arrowhead of the sender), and
2. what transition ("tilstandsovergang") does it execute next?

Scenario	Message Sequence
a)	 Sender Receiver rdt_send(data) (0,data) deliver(data) (0,ACK)
b)	 Sender Receiver rdt_send(data) (0,data) deliver(data) (0,ACK)

c)	Sender Receiver rdt_send(data) (0,data) deliver(data) (0,ACK) rdt_send(data) (1,data)
d)	Sender Receiver rdt_send(data) (0,data) deliver(data) (0,ACK) rdt_send(data) (1,data) deliver(data)

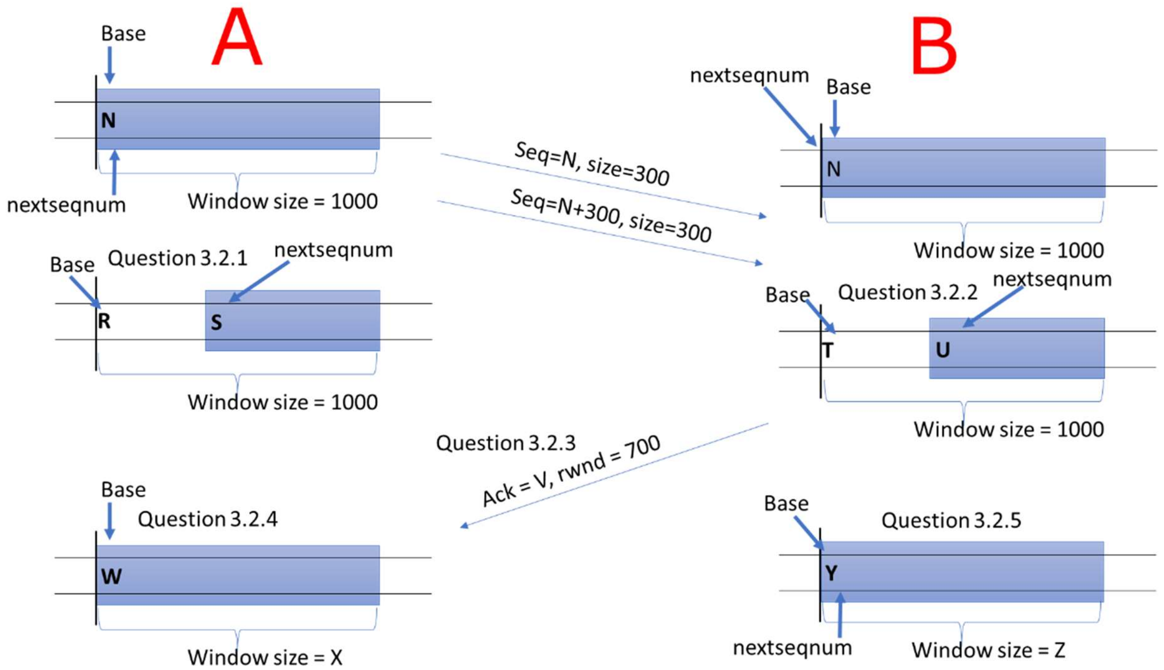


ANSWER		
Scenario	Sender State	Next sender transition
a)		
b)		
c)		
d)		

Question 3.2 (12 pts).

Let us consider a TCP 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 in bytes. An ACK packet specifies what data bytes is being ACKed, and it can set the new size of the TCP window.

Consider next figure, where both *base* and *nextseqnum* are represented as N. To solve the exercise, consider as a concrete value for N the last 4 digits of your student number. For example, if your student number is 203456, consider N equal to 3456. Compute the values of R, S, T, U, V, W, X, Y, Z to answer the questions below.



Q 3.3.1: After A sent two segments of 300 bytes each, what are Base R and "nextseqnum" S of A's window?

Answer	N= R= S=

Q 3.3.2: After B received both fragments, what are Base T "nextseqnum" U of B's window?

Answer	T= U=

Q 3.3.3: Consider that B sends an acknowledgment message after it received both messages from A, and sends back a rwnd of 700. What is the value of the ACK field V of the ACK message?

Answer	V=

Q 3.3.4: Please complete the window of A with base/nextseqnum W and window size X after receiving the ACK:

Answer	W= X=

Q 3.3.5: Please complete the window of B with base/nextseqnum Y and window size Z after having sent the ACK:

Answer	Y= Z=

4. Network Security (8 pts)

Question 4.1 (4 pts)

What properties allow the attacker Trudy to perform a Man-in-the-Middle attack to the communication between Alice and Bob? Choose all that applies.

1. Trudy knows the private key of Alice prior to the communication
2. Trudy is able to put herself on the route between Alice and Bob.
3. Alice and Bob do not know each other before communicating e.g.: they did not exchange encryption keys before starting the encrypted communication.
4. The communication is using TCP as the transport protocol, which is inherently insecure

Answer	

Question 4.2 (4 pts)

Consider that you just received a certificate. In which case can you trust the certificate for secure SSL communication? Choose all that applies.

1. The public key on the certificate contains enough bits to be considered secure
2. The certificate was sent you by email by a friend you trust
3. The certificate was pre-installed on your browser, and its certificate is issued by an official certification authority
4. The certificate is signed by a certification authority you trust, and you can verify the signature

Answer	

5. Practical Assignment (33 pts)

A group of friends like tasking beers. Being grounded at home, they would like a web-based app to help them share their tasting impressions. Your task is to help them by developing a simple beer tasting system.

The app should have two main functionalities:

1. Allow a user (identified by name) to report a score of a beer.
2. Allow the user to show the scoreboard of the beers: a table of beers sorted in the order of highest average score.

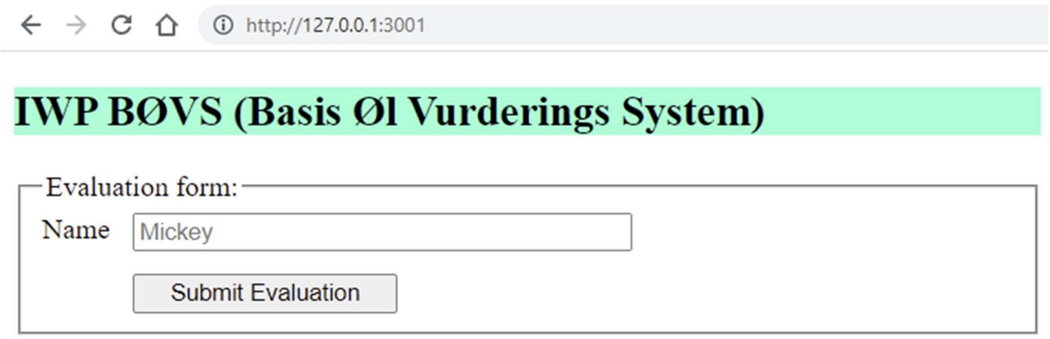
The file (download appendix from digital exam) `NodeWeb-BOVS.zip` contains a skeleton for this application. Unzip the archive to a suitable location on your machine. A workspace for visual studio code is stored in the file `bovs.code-workspace`. The goal of the assignment is to add the required functionality to the skeleton.

A small node.js application (the main application file `app.js` and file `server.js` with helper functions to serve files and json data) acts as a http server that serves the front-page (stored in `bovs.html`) and a web-API function skeleton for storing JSON objects containing the evaluation information for beers.

The beer score information is stored in an in-memory "database" implemented as a simple (`beerScoresDB`) array of records. A record is a JavaScript object with the following properties *evaluatorName, beerName, score.* In the first version that you develop here, the list of candidate peers is static, and a beer simply identified by its name as a string.

In the current state, the server accepts an HTTP POST to the resource name `/beerEvaluations` that is to store a new record with the evaluation information that the user has entered.

When the server is started (either run the app.js from within VisualStudio Code, or the command line NodeWeb-BOVS >`node node/app.js`) and a web browser is pointed to the localhost at port 3001, the browser should show the following page[1]:



http://127.0.0.1:3001

IWP BØVS (Basis Øl Vurderings System)

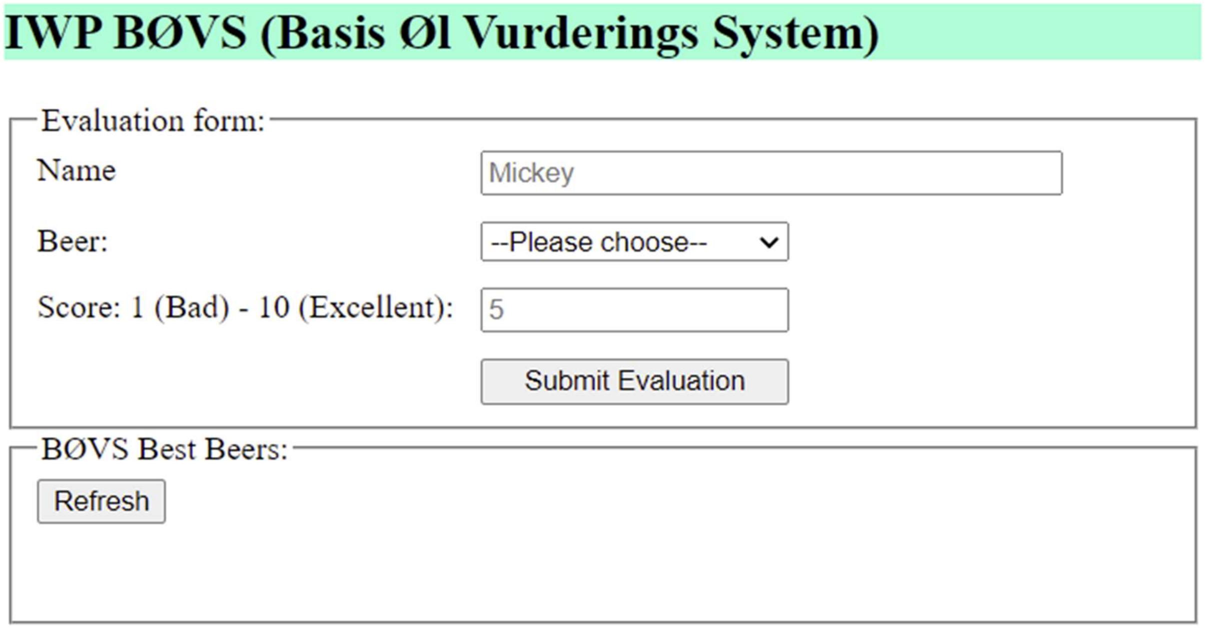
Evaluation form:

Name Mickey

Submit Evaluation

[1] BEWARE: The supplied code uses ECMASCRIPT modules; hence use the supplied workspace and the package.json file in the supplied archieve. This file sets the required 'type="module"' setting.

At the end of the assignment your frontpage should appear somewhat like the screenshots below: In the first part reports to the app that the named person has conducted an exercise event with a given number of steps. The second part lists the current scoreboard whenever the user refreshes it. Focus on correct and functional HTML and JS. **DO NOT SPEND TIME ON LAYOUT**, but optionally you may use the supplied css-file.



IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

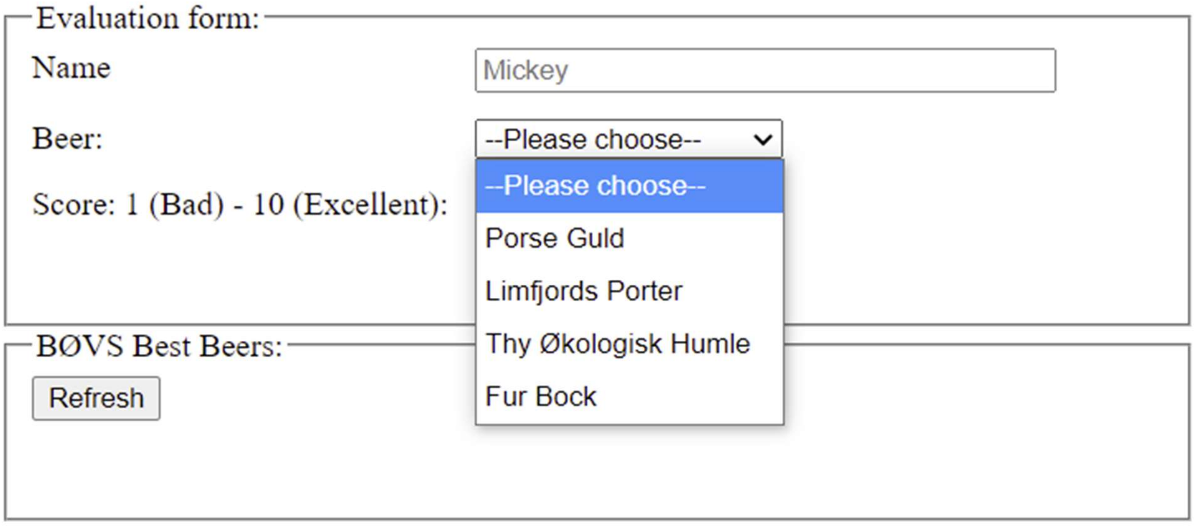
Name: Mickey

Beer: --Please choose--

Score: 1 (Bad) - 10 (Excellent): 5

BØVS Best Beers:

Before refreshing scoreboard



Evaluation form:

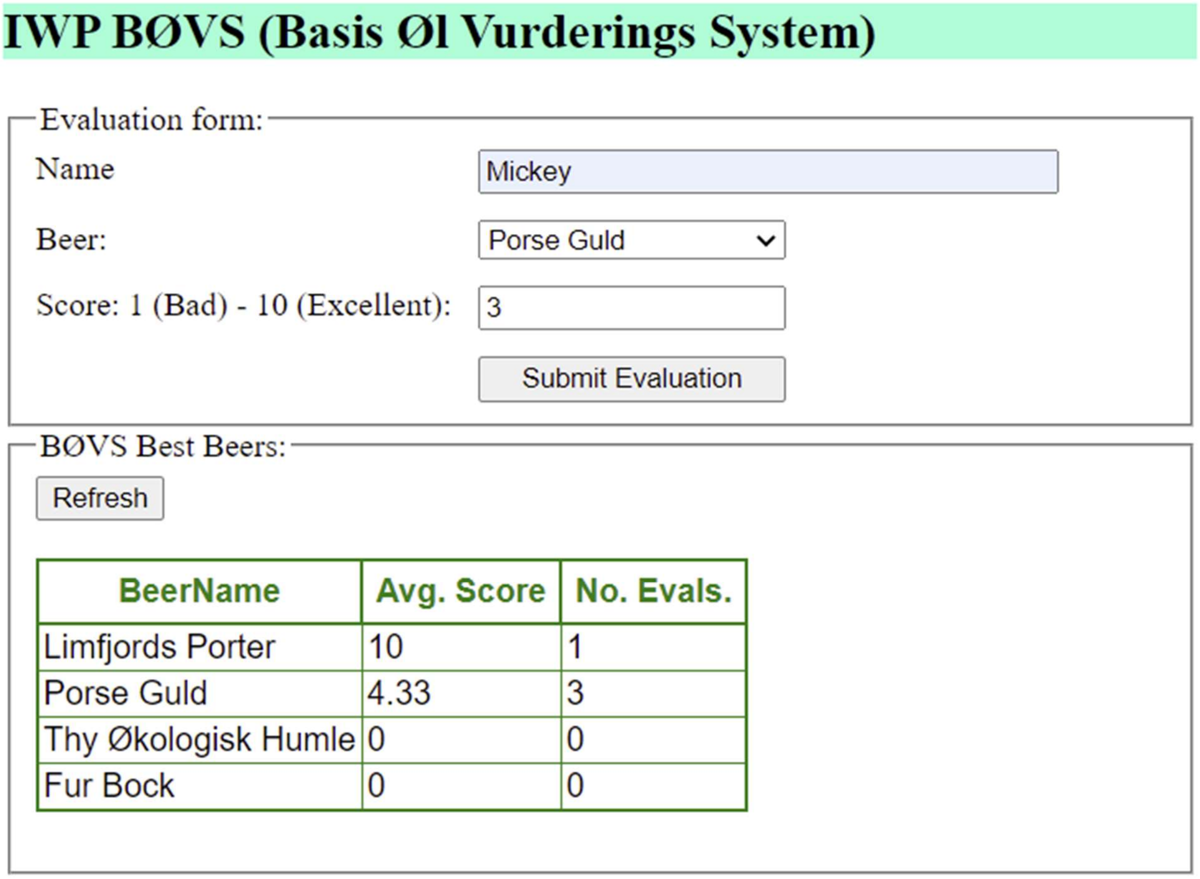
Name: Mickey

Beer: --Please choose--

Score: 1 (Bad) - 10 (Excellent):

BØVS Best Beers:

Before refreshing scoreboard, showing the static list of beers



IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

Name: Mickey

Beer: Porse Guld

Score: 1 (Bad) - 10 (Excellent): 3

BØVS Best Beers:

BeerName	Avg. Score	No. Evals.
Limfjords Porter	10	1
Porse Guld	4.33	3
Thy Økologisk Humle	0	0
Fur Bock	0	0

After refreshing scoreboard

Question 5.1: (6 pts)

Extend the evaluation form with a menu to select a beer to evaluate, and a field to enter the score. Use *HTML validation* to ensure that they are filled out and satisfy the constraints: 1<=score<=10. ***Add*** the form for the scoreboard part. In an addition to the refresh button, it should include a field for the scoreboard output. The contents of that field is to be populated later in the exercise.

Question 5.2 (4 pts)

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

Question 5.3 (5 pts)

At the server `app.js`, a POST to the resource `/beerEvaluations` results in that the received JSON object is converted into a JavaScript object with the new submitted evaluation form data, validated, and then conditionally inserted in the “database”. Show the updated validation function.

Question 5.4 (5 pts)

Implement a JavaScript function `calcHighScores` that computes and returns a scoreboard object. The scoreboard, like the database, is an array of java objects each representing a record with a beer-name field, average score, and number of evaluations. If you wish to skip this question for now, simply return a small manually array with fixed dummy values).

Hint: One way to compute the scoreboard is to iterate though the possible beers (`beersDB`) and sum the evaluation scores for each beer. When one beer is completed, the record can be added to a separate result array. At the end, this array can be sorted by average score. Keep in mind that some beers may not have received any evaluations.

Question 5.5 (5 pts)

Extend (the server’s web-api) such that a HTTP GET call to an endpoint (resource name) of your own choice returns a JSON object to the client containing the current scoreboard. Show here your code for the route-handling, request processing, and possible extra functions that you have added.

Question 5.6 (6 pts)

Implement the client-side functionality for the scoreboard. I.e., when the user clicks refresh, it should fetch the current scoreboard, and update the HTML element that is to contain the result. For full points, format the scoreboard as a HTML5 table (if out of time, show the json object.). Only 1 table should appear. **Additionally**, add an entry to the console log containing the highScore table, when a new response is received.

HINT: A helper function to fetch json objects is provided.

Question 5.7 (2 pts)

Show the client-side log contents resulting from submitting a beer evaluation, and refreshing the scoreboard (a snip-it/screendump) image is fine.