

IWP: Internetwork- and web-programming

Written exam, August 30, 2021. Aalborg University

- You must upload your 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 (19 pts)	2
2. Computer Networks and the Application layer (15 pts)	5
3. Reliable Data Transfer and the Transport Layer (25 pts)	7
4. Network Security (8 pts)	12
5. Practical Assignment (33 pts)	13

Michele Albano and Brian Nielsen

1. HTML and HTTP (19 pts)

Question 1.1 (9 pts)

Q1.1.1 Which of the elements below does not belong to the html document header?

1. TITLE
2. HEADER
3. SCRIPT
4. META

Answer	Header
--------	--------

Q1.1.2 Which purposes does the attribute “id” serve on an HTML element? State all correct answers.

1. Generates a key/value pair consisting of the given id and the value of the element
2. Allows a stylesheet to identify the element
3. It makes it possible for the user to input a unique id into the element
4. Allows a script to lookup the element

Answer	2 4
--------	-----

Q1.1.3 Separating style from semantic markup of an HTML document into a “cascading stylesheet” file is beneficial because

1. It is the only permitted method in html5
2. It is easier to maintain as style changes only needs to be made to a single location.
3. The browser renders the page faster.
4. People with eye disabilities can use an alternative style

Answer	2
--------	---

Q1.1.4 What is the correct HTML syntax for creating a sentence of text: “My first computer had 1 kb RAM”, where “first” is marked with a hyperlink to the destination “zx81.dk”, to be accessed using the secured hypertext transport protocol.

Answer	<p> My first computer had 1 kb RAM </p>
--------	---

Q1.1.5 What output does the following JavaScript program fragment output to the console when the requested server is unavailable/down?

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

Answer	Start, Failure!, Finished
--------	---------------------------

Q1.1.6 Consider the `fs.readFile(path[, options], callback)` of the `fs` module. Which of the following are true?

1. The `readFile` function will block program execution for a long time if the file being read is very large
2. The `readFile` function returns a Promise
3. The “callback” function will be executed when the file is read
4. It is possible to wrap the execution of `readFile` into a Promise (for example using the `Promisify` function)
5. To access the file being read, it is necessary to invoke the “`then()`” function on the result of the `readFile` call
6. It is recommended to use the `readFile` function only on files in the JSON format

Answer	3 1 4
--------	-------

Question 1.2 (5 pts)

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

```
GET /suggestion/k/klassefesten.json HTTP/1.1\r\n
Host: v2.sg.media-imdb.com\r\n
Connection: keep-alive\r\n
sec-ch-ua: "Chromium";v="92", " Not A;Brand";v="99", "Microsoft Edge";v="92"\r\n
Accept: application/json, text/plain, */* 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/92.0.4515.131 Safari/537.36 Edg/92.0.902.67\r\n
Origin: https://www.imdb.com\r\n
Sec-Fetch-Site: cross-site\r\n
Sec-Fetch-Mode: cors\r\n
Sec-Fetch-Dest: empty\r\n
Referer: https://www.imdb.com/\r\n
Accept-Encoding: gzip, deflate, br\r\n
Accept-Language: da,en;q=0.9,en-GB;q=0.8,en-US;q=0.7\r\n
\r\n
```

```
HTTP/1.1 200 OK\r\n
Content-Type: application/json\r\n
Content-Length: 905\r\n
Connection: keep-alive\r\n
Server: Server\r\n
Date: Fri, 13 Aug 2021 11:26:43 GMT\r\n
x-amz-rid: AKM3YZWT0WA1EQK76A4D\r\n
x-amzn-RequestId: a5792154-8c96-4b2c-aa85-89fcd396d470\r\n
Access-Control-Allow-Origin: https://www.imdb.com\r\n
Content-Encoding: gzip\r\n
Access-Control-Expose-Headers: x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date,link\r\n
```

```

Link: </schemas/json/response/imdb/api/search/suggestion/response.js>;
rel="describedby"\r\n
Permissions-Policy: interest-cohort=()\r\n
Via: 1.1 e8b17f734954ee4d46d26cf302323482.cloudfront.net (CloudFront), 1.1
91dfa0ccf808a1c0b41f43736ca93f80.cloudfront.net (CloudFront)\r\n
X-Amz-Cf-Pop: FRA53-C1\r\n
Vary: Origin,Accept-Encoding\r\n
X-Cache: Hit from cloudfront\r\n
X-Amz-Cf-Pop: CPH50-C2\r\n
X-Amz-Cf-Id: hFXB-gFOSevKbulsbpb6V2ZntykGr61pYbz1xc8G-A5wNbHnk3LMSQ== Age: 255\r\n
\r\n

```

Question	Answer
1. Which http method is used?	GET
2. Which protocol is used?	HTTP 1.1
3. What resourcepath is accessed at the server	/suggestion/k/klassefesten.json
4. Which protocol and server was the origin of the request?	IMDB.com 1.1
5. How is the response encoded?	gzip

Question 1.3 (5 pts)

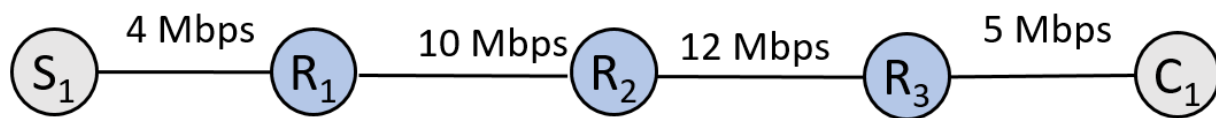
With regards to the HTTP methods, which HTTP method(s) are better in the following situations?

Situation	Method(s) (GET/PUT/POST/DELETE)
The goal is to update an existing resource	Put
A new resource must be created, even if a resource already exists	Post
The goal is to execute an idempotent change on the resource	Put
If the resource exists, it must be deleted	Delete
The operation must retrieve data from a resource	Get

2. Computer Networks and the Application layer (15 pts)

Question 2.1 (4 pts)

Consider the network shown in the figure below that shows 1 server (S_1) and 1 client (C_1). The network is interconnected using the routers ($R_1 \dots R_3$) with the possible transmission rates for each link. The client is to download a file of 1G bits size. Assuming no other traffic on the network and a 20% overhead for protocol headers, how long time does it take the client to download the file?



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

Answer	$1\text{Gbit} * 1.2 = 1.2\text{Gbit} * 1000 = 1200\text{Mbits} / 4\text{Mbps} = 300\text{ Sec}$
--------	---

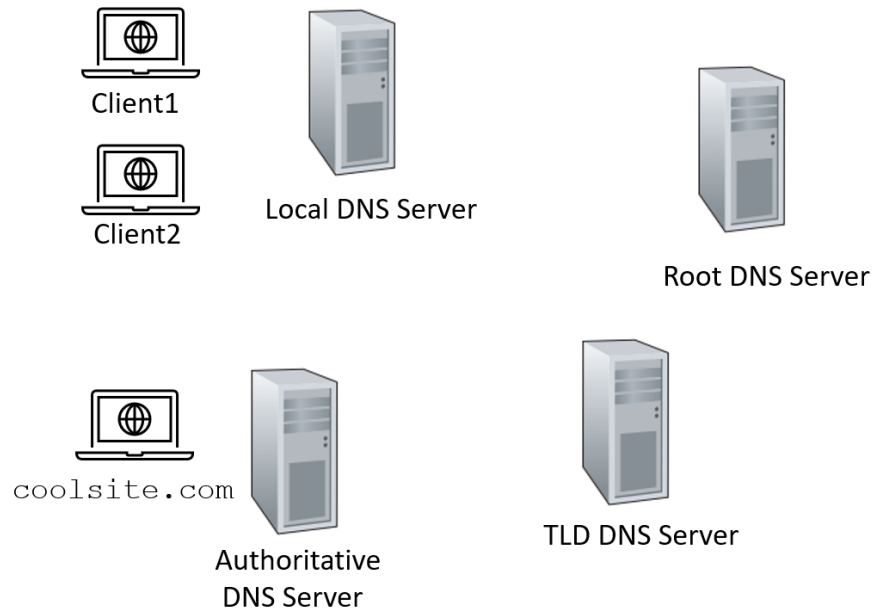
Question 2.2 (5 pts)

On what layers in the TCP/IP protocol stack does the following activities take place?

Activities	Name of responsible layer
File Transfer	Applikation
Remote Login / remote desktop	Applikation
Computing UDP checksum	Transport
Transferring a data-frame from a client to a router	Netværk
Transferring a data-frame on Ethernet-cable	Link

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 a **recursive** 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:

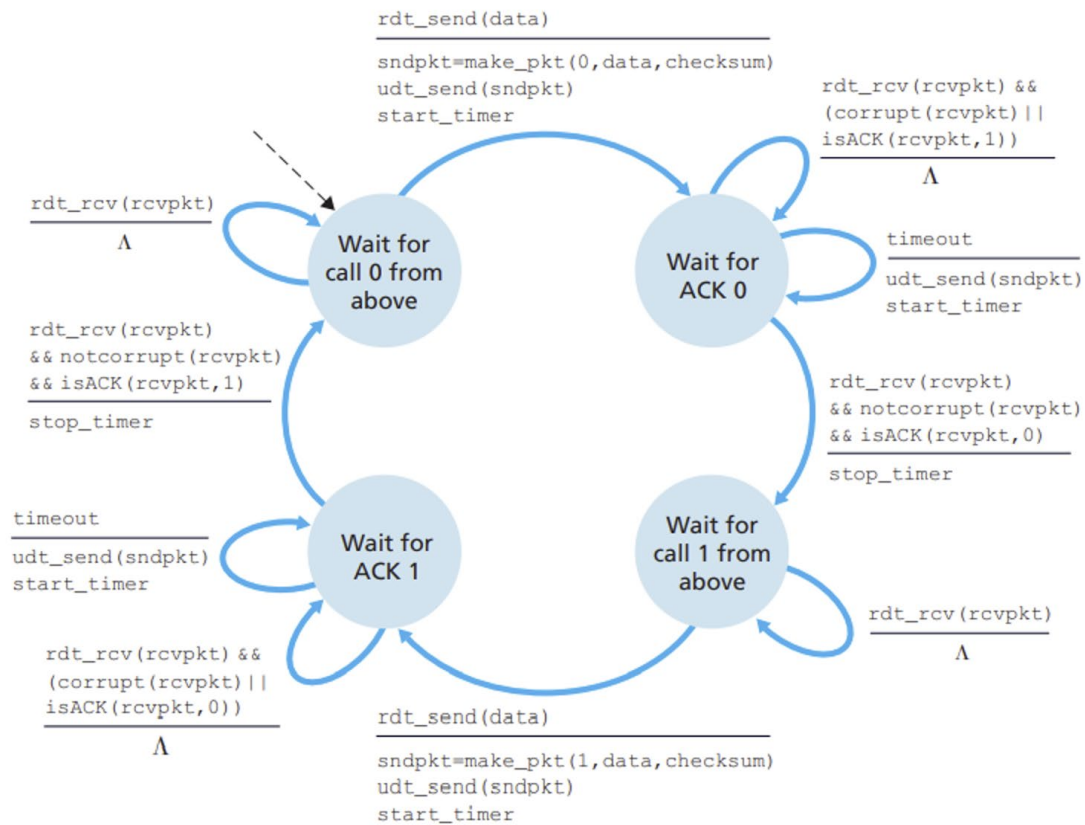
- A->B if host A sends a request to host B.
- A<-B if host B sends a reply to host A
- A<-> if host A sends a request to host B which immediately replies to A
- Sequence can be indicated by numbering or semi-colon: e.g: A->B; B->C

Answer 1	Client 1 ->Local DNS Server ->Root DNS Server -> TLD DNS Server <-> Authoritative DNS Server Client 1 <- Local DNS Server <- Root DNS Server <- TLD DNS Server
Answer 2	Client 2 -> Local DNS Server <-> Root DNS Server Local <-> TLD DNS Server Local <-> Authoritative DNS Server Client 2 <- Local

3. Reliable Data Transfer and the Transport Layer (25 pts)

Question 3.1 (6 pts)

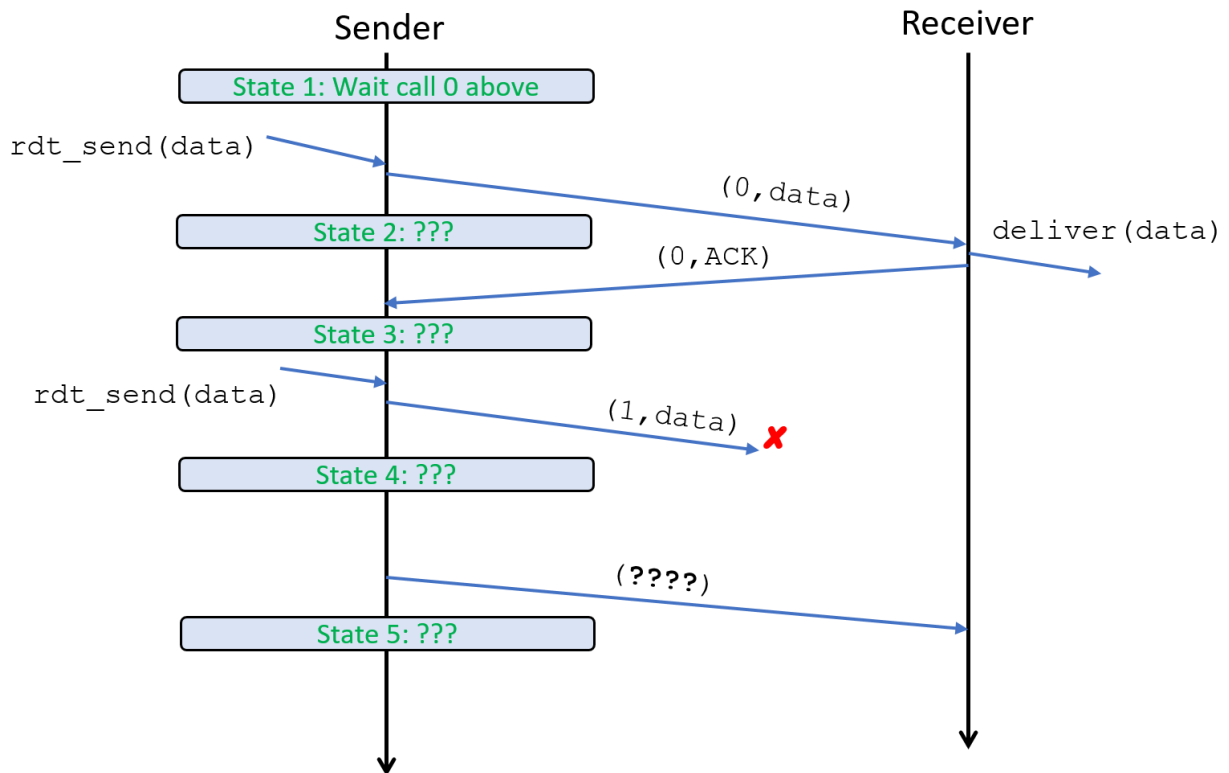
Two hosts use the alternating bit protocol (rdt3.0) to reliably transfer data. Consider the sender protocol state-machine in the below figure (adopted from Kurose&Ross Fig 3.15):



Question 3.1.1 (3 pts)

Consider the sequence diagram below that have been extended to show the state-name resulting after each protocol event (in the blue-shaded rectangles with rounded corners). Complete the diagram with the state-name for the missing states.

Also, what is the contents of the message labelled with question marks?

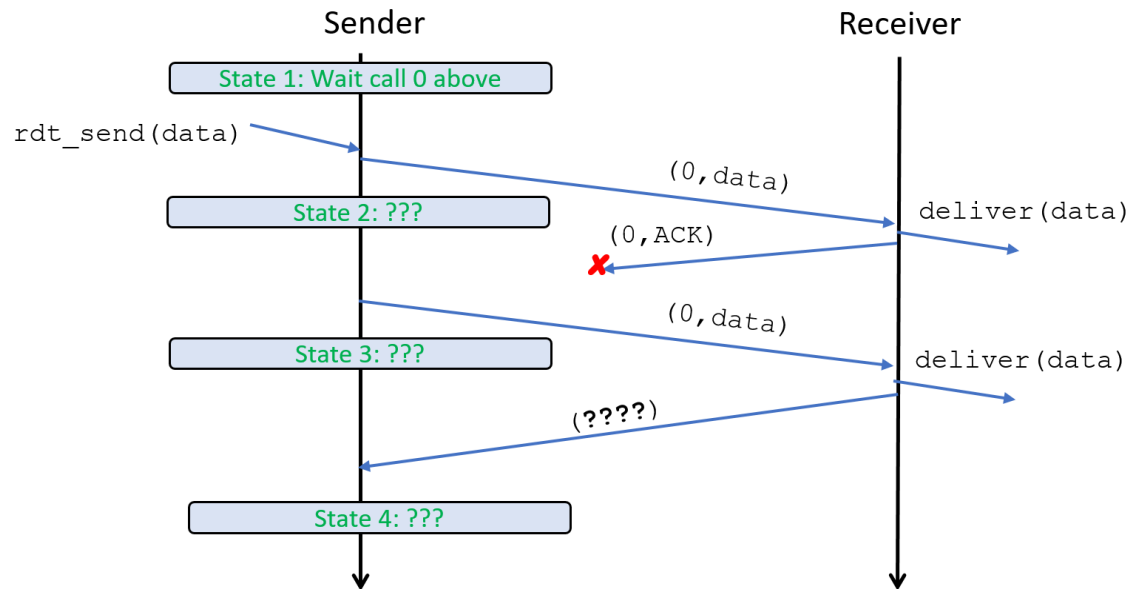


ANSWER	
State 1	Wait call 0 from above
State 2	Wait for ack 0
State 3	wait call 1 from above
State 4	Wait for ack 1
State 5	Wait call 1 from above
Message contents	1, data

Question 3.1.2 (3 pts)

Consider the sequence diagram below that have been extended to show the state-name resulting after each protocol event (in the blue-shaded rectangles with rounded corners). Complete the diagram with the state-name for the missing states.

Also, what is the contents of the message labelled with question marks?

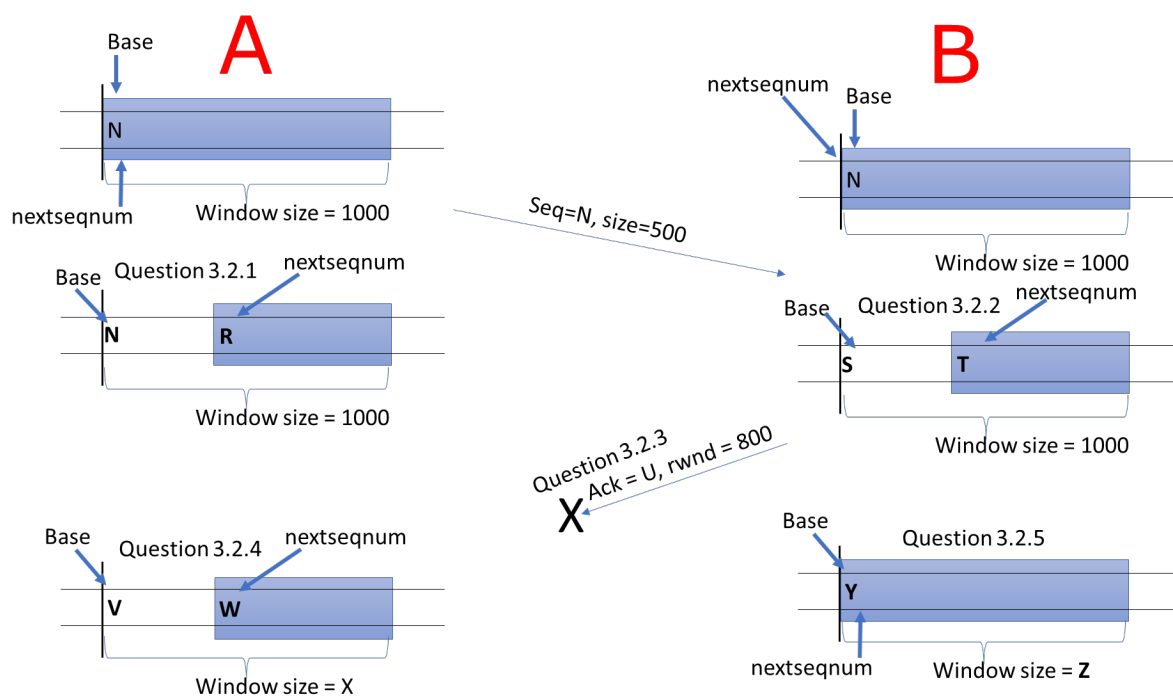


ANSWER	
State 1	Wait call 0 from above
State 2	Wait for ack 0
State 3	Wait for ack 0
State 4	Wait call 1 from above
Message contents	0, ACK

Question 3.2 (12 pts).

Let us consider a TCP exchange of segments between A and B, where the acknowledgment gets 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.

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 3.2.1: After A sent one segment of 500 bytes, what is “nextseqnum” of A’s window?

Answer	R = 3956
--------	----------

Q 3.2.2: After B received the fragment, what are Base and “nextseqnum” of B’s window?

Answer	S = 3456 T = 3956
--------	----------------------

Q 3.2.3: Consider that B sends an acknowledgment message after it received the message from A, and sends back a rwnd of 800. What is the value of the ACK field of the ACK message?

Answer	U = 3956
--------	----------

Q 3.2.4: Consider that the acknowledgement message from B was lost. Please complete the window of A with base, nextseqnum and window size after that event:

Answer	V = 3456 W = 3956 X = 1000
--------	----------------------------------

Q 3.2.5: Please complete the window of B with base/nextseqnum and window size after having sent the ACK:

Answer	Y = 3956 Z = 800
--------	---------------------

Question 3.3 (7 pts)

Of the following clauses, say which ones apply to the rwnd (receive window), and say which ones to the cwnd (congestion window). Mark each cell by true or false.

Question	RWIND	CWIND
1) It is used to limit how much data can be sent before an acknowledgment	F	T
2) It uses heuristics since it depends on information that cannot be fully known	T	T
3) Can be set up by the host receiving and acknowledging the data	T	F
4) It can be used on encrypted connections only	F	F
5) It is measured in bytes	T	T

4. Network Security (8 pts)

Question 4.1 (4 pts)

Which ones of the following ensures “message integrity”?

- 1) Sending a copy of the message using a protected channel, and then comparing it with the received message
- 2) Sending multiple (at least 3) copies of the message
- 3) Assuming the receiver knows the public key of the sender, signing the message with the private key of the sender.
- 4) Assuming the receiver knows the public key of the sender, encrypting the sent message with the private key of the sender,
- 5) Signing the message with the public key of the receiver, since the receiver is the only one knowing its own private key

Answer	3
--------	---

Question 4.2 (4 pts)

Which of the following are among the goal of the SSL handshake?

- 1) To select the symmetric encryption algorithm for the confidentiality of the communication
- 2) To select the MAC algorithm to ensure the integrity of the messages
- 3) To generate the key used for encrypting the communication
- 4) To authenticate the server with the client
- 5) To authenticate the client with the intermediate routers

Answer	1 3 4
--------	-------

5. Practical Assignment (33 pts)

A group of friends like tasting 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 a (possibly other) user to create a new beer to be evaluated.

The file (download appendix from digital exam) `NodeWeb-BOVS2.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 `bovs2.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 list of possible beers that can be evaluated is stored in an in-memory “database” implemented as a simple array of (`beersDB`) records. A record is a JavaScript object with the following properties *creatorName*, *beerName*.

Similarly, the beer score information is stored in an in-memory “database” implemented as a simple (`beerScoresDB`) array of records: *evaluatorName*, *beerName*, *score*.

In the current state, only a single, statically defined, beer can be evaluated. A beer simply identified by its name as a string. Also, the server accepts an HTTP POST to the resource name `/beerEvaluations` that stores 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-BOVS2 > node node/app.js`) and a web browser is pointed to the localhost at port 3001, the browser should show the following page¹:

¹ BEWARE: The supplied code uses **ECMAScript** modules; hence use the supplied workspace and the package.json file in the supplied archive. This file sets the required ‘type=“module”’ setting. IF you cannot run the program due to a module error try the following:

1. copy and paste the contents of `server.js` into `app.js`
2. comment any line starting with `import/export`
3. uncomment the lines starting containing the “require” statements
4. remove the line from `package.json` that sets module type (`"type": "module"`)

IWP BØVS (Basis Øl Vurderings System)

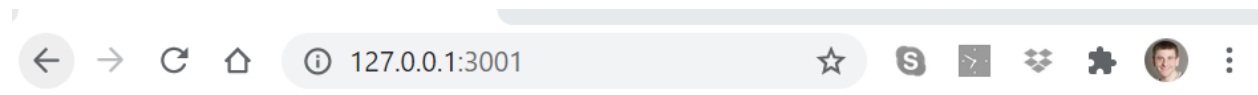
IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

Name	<input type="text" value="Mickey"/>
Beer:	<div>--Please choose-- ▾</div>
Score: 1 (Bad) - 10 (Excellent):	<div>--Please choose-- Porse Guld</div>
<input type="button" value="Submit Evaluation"/>	

At the end of the assignment your frontpage should appear somewhat like the screenshots below: Your task is firstly to extend this skeleton with a section that allows a user to create a new beer that can be evaluated. Secondly, your task is to allow the new beers to be evaluated.

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)

IWP BØVS Admin

Create new beer:

Creator Name	<input type="text" value="Mickey"/>
New Beer Name	<input type="text" value="Beer name"/>
<input type="button" value="Create Beer"/>	

IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

Name	<input type="text" value="Mickey"/>
Beer:	<input type="text" value="--Please choose--"/>
Score: 1 (Bad) - 10 (Excellent):	<input type="text" value="5"/>
<input type="button" value="Submit Evaluation"/>	

Before creating a beer.

← → ↻ 🏠 ⓘ 127.0.0.1:3001 ☆ S 🔍 📁 ⚙️ 👤 ⋮

IWP B

IWP BØ

127.0.0.1:3001 says

Ok! Beer Created.

OK

Create new beer:

Creator Name

Mickey

New Beer Name

Tuborg

Create Beer

IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

Name

Mickey

Beer:

--Please choose--

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

5

Submit Evaluation

After creating a new beer

IWP BØVS (Basis Øl Vurderings System)

IWP BØVS Admin

Create new beer:

Creator Name

New Beer Name

IWP BØVS (Basis Øl Vurderings System)

Evaluation form:

Name

Beer:

--Please choose- ▾

--Please choose-
Porse Guld
Limfjords Porter
Thy Økologisk Humle
Fur Bock
Tuborg

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

The choice-list of possible beers should updated from the server to **include newly created beers**.

Question 5.1 (5 pts)

Extend the front page with a section and form allowing a beer to be created. It should have two fields: the name of the person creating the beer, and the name of the beer. Use *HTML validation* to ensure that they are filled out and name and beer lengths are between 1 and 30.

Question 5.2 (3 pts)

Make a JavaScript function at the client that is able to extract creator name and beer name from the form. The function is to be called when the forms submit ("create") button clicked (next exercises). Log the extracted data to the console.

Hint: You may want to take a look at the client code to see how the evaluation-form is handled

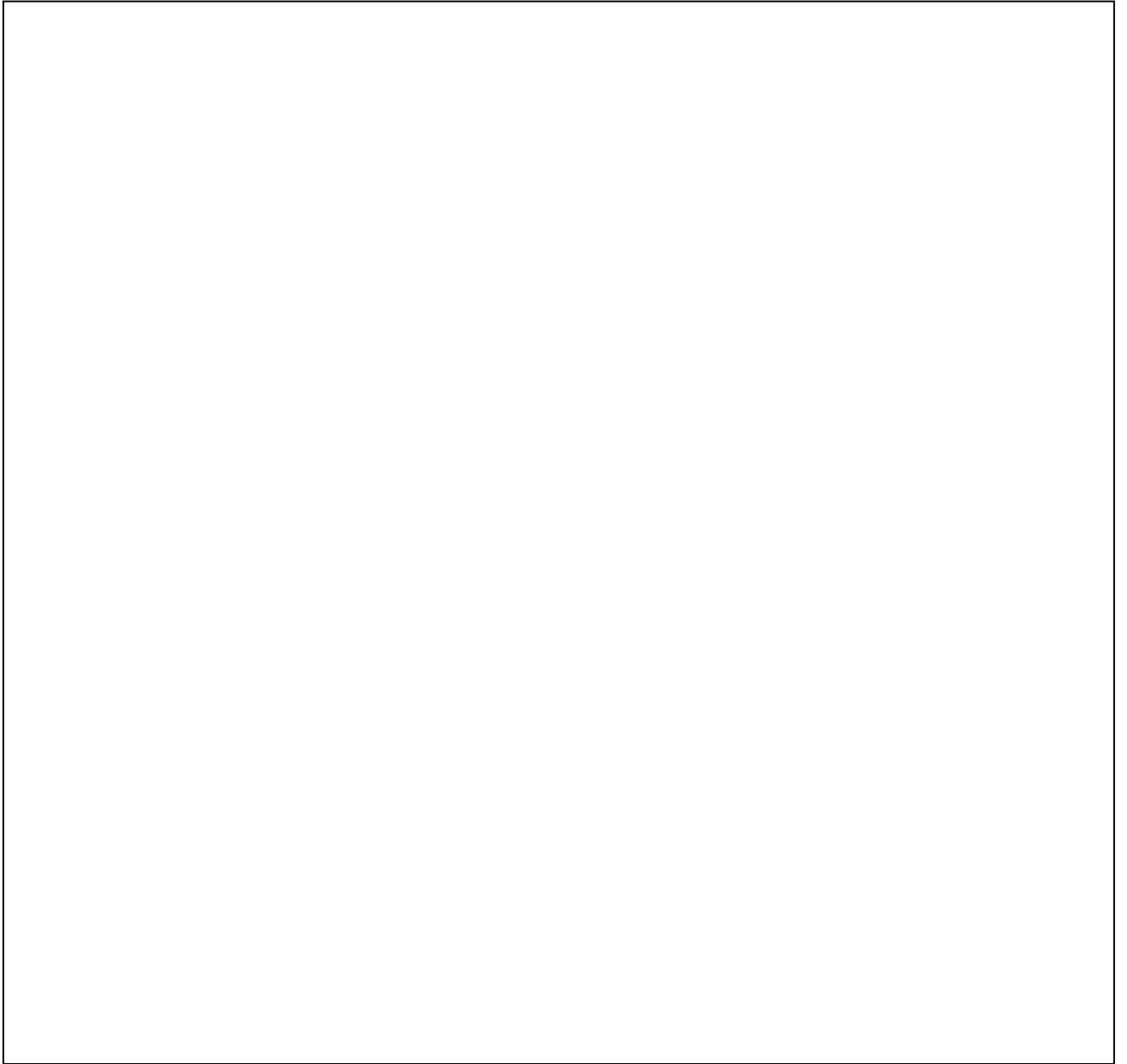
Question 5.3 (5 pts)

To create a beer, the server offers a resource: `/beers`, that clients can use to POST a json object containing two string fields `"creatorName"` and `"beerName"`. As result, the POST will return a json encoded Boolean that indicates whether or not the beer could successfully created in the server-side database.

Write a function at the client that:

1. Uses the extract function from the previous exercise to extract form data.
2. Makes an HTTP POST of the extracted object to the server resource `/beers`
3. Log the result of the operation to the console.
4. The result of the operation should show an alert-box, that shows:
 - 1) *"Ok! Beer Created"* in case of success
 - 2) *"Cannot Create! Beer exists."* in case of non-success
5. Finally, the function should be registered as event-handler and called when the user submits the form; i.e. your java-script should handle the server communication, not the browsers default forms-handling.

Hint: You may want to take a look at the client code to see how the evaluation-form is handled



Question 5.4 (4 pts)

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

Question 5.5 (4 pts)

Implement a JavaScript function `getBeers` that computes and return an array of the currently registered beer names in the `beers-database`.

Question 5.6 (4 pts)

Extend (the server's web-api) such that a HTTP GET call to /beers returns a JSON formatted array of beer-names (use the function from the previous question). Show here your code for the route-handling, request processing, and possible extra functions that you have added.

Question 5.7 (6 pts)

Write a client-side function that retrieves the currently registered list of beers at the server (i.e. that may recently have been added by other users of the system), and updates the beer-select menu of the evaluation-form to contain the retrieved list of beers. The selection option text and option value should both be set to the beer name. Log the retrieved list to the console log.

Setup an interval timer such that the select menu is updated every 30 seconds.

HINT: A helper function to fetch json objects is provided. Also, the function `element.replaceChildren()` may be usefull. With an empty argument list, it deletes the children of the element.

Question 5.8 (2 pts)

Show the client-side log contents resulting from creating a new beer, and after a while evaluating it. A snip-it/screendump) image is fine.

A large, empty rectangular box with a thin black border, intended for a user to paste a screenshot of client-side log contents. The box occupies the majority of the lower half of the page.