CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT 341	WEB & MOBILE DEVELOPMENT			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
894		2020-01-07	9	C

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	x	24	
2	x	6	
3	x	14	
4	x	11	
5	x	6	
6	x	20	
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		81	

[redacted]

ıe examinee
e examina-

e University

he equip-
uipment not

mber, name
thing has

l photo ID
iving licence.
sn't have a

ng language
iiner permits

demic year.

ulators.

vith this

o the different

ıe senior
ll decide

o discard old

.aspx

3

(1)

Q1.1: var - is functionscoped (value available for the function it is specified in)
let - is block scoped
const - is a final value that is assigned, it cannot be changed

Q1.2: METHOD / Protocol (version) /path
Host:
Content-type: } Header
Body: } Body

GET / HTTP 1.1 / 'names'
Host: Google.com
Content-type: HTML/XML

Q1.3: <p> Stands for paragraph, this is used when writing shorter or longer texts under headings.

 This sections different parts of the html, scopes it so that you can easily attach an id or class to it, which makes modifying easy.

 is used when you want to alter a shorter line or a few words in a text to change its color or whether its bold or not (examples). ✗ repeat on styling

Q1.4: p ~ span stands for "all siblings of p which are spans" (in my words). Hence all the attributes after a <p> is closed will adhere to any constraints set. ✗ on the same level!

Q1.5: AJAX = Asynchronous Javascript and XML
- Allows for specific information to be rendered after a page has already been loaded (without reloading) - automatically updates
- JS and text (XML, JSON) are combined
- other HTTP requests would have to reload the page
- Uses a library, such as axios, for the requests.

Q1:6: HATEOAS = Hypermedia as the engine of application State

It states that the user knows only one entry-point. Everything else it set as hyperlinks within the page that redirects the user.

Q1:7: When implementing two-way data binding, not only does changing the model affect the view, but modifications in the view also changes the model through the Vue instance, which acts as the view-model. One can use the v-model infront of attributes to recieve this.

Q1:8: Process = It is a running program / independent sequences of execution - it has allocated memory - it cannot communicate with other processes space

Thread = It is an entity within a process.
- runs in parallel with other threads
- can communicate with other threads
- Shares memory space and stack.

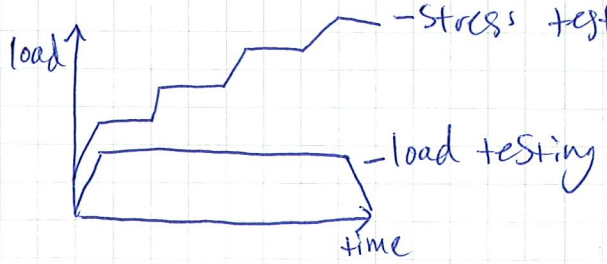
Q1:9: Domain Name System changes IP addresses into human readable names.

ex) 172.253.... = www.google.com

Names are given based on your IP address.

Q1:10: Load testing is done by testing a heavy load under a longer amount of time to see how the application handles it.

Stress testing timingly increases the load in intervals to see how the application handles it.



Q1.11: Infrastructure as a Service (IaaS)

This model provides the user with VMs or containers. Although, the developer must then himself/herself implement/develop servers and the application

Platform as a Service (PaaS)

This model provides for runtime environments, databases and application servers, along with the VMs. The developer can then focus on creating the application and not how to handle the data

Software as a Service (SaaS)

This model has the two services above provided as well as an application. This would leave only minimal customization for the developer.

Example Services 4

3

(2)

```
18  <h1> DIT341 Table Example </h1>
19  <h2> An example for a table in HTML: </h2>
20  <table summary = "looked -at()">
21  <th> Examination
22      <tr> Exam </tr>
23  </th>
24  <th> Type
25      <tr> Written </tr>
26  </th>
27  </table>
28  <p> <span class = "left"> This is written left. </span>
29
30  <span class = "right"> And this is right. </span>
31  </p>
```

161

6

3

(3)

13 class = "blue
14 id = "red"
15
16 –
17 –
18
19 –
20 –
21
22
23 div
24 custom = "pink"
25 div

4

14

3

(4)

```
9   <list v-for="item in items">
10  <ul>
11  {{ item }}
12  </ul>
13  </list>
14  <button onClick="addToList(to_add)> Button </button>
    :
26  addToList(to_add) {
27  items.push(to_add)
28  }
```

17

3

(5)

- There are two textboxes (just text) set on the page: txtView Chalmers and txtView GU

  At the bottom request1 and request2 are added to the request queue called queue. In turn, a GET request is invoked. A ~~GET~~ request along with the first url is sent to the backend and the response given back will replace the ~~text view~~ txtView Chalmers.

  The same goes for request2. The GET request is sent to the backend but with a different URL. The response is set as a text in the original textbox = txtView GU

- If successful, code 200 is shown and the response replaces the text in the textbox (if there is any). If one is successful and the other is unsuccessful, one will have a response and the other one will not. They do not depend on each other, unless there is a server error which may cause the app to crash.

- The txtViewChalmers is filled first.
  The txtView GU is filled second.

6

Q6:1: GET /:user/documents

We would receive all the documents for a specific user (user) ~~access~~ If successful, a return code of 200 would be shown, as ok request.

POST /:user/documents

This would create a new document to a specific user. If successful, a status code of 201 would be shown, as created.

Q6:2: GET / HTTP 1.1 / ~~user~~ /documents ~~.html~~
Host: www.googledocs.com
Content-type: HTML

Q6:3: 401: unauthorized
404: not found

If a user has been logged out or in general does not have the permission to access or edit that document, 401 will be shown stating that the current user is unauthorized for the given request.
Ex. in google docs you can have a view function with not being allowed to edit.

If a user does not have the document remaining, a status code of 404 would be displayed, indicating it is not found.
Ex. When sharing documents online where others can edit it, there can happen that what you want to update no longer exists.

3

Q6:4: NO, beause it is not very clear, also why would you post something that has to be deleted? Instead of using POST, one can use DELETE and remove the last path. DELETE /:user/documents/:doc, and you are set! ✓

Q6:5: This operation is trying to edit or update a specific document that belongs to a specific user. The user would then have a list of documents when logging in (probably) and this request would edit one of those documents.

Depending on what the user is wanting to update, those fields will be changed in the document, and the rest will stay the same. 3

2. Parameters could be things such as an email and password in order to access the documents, then you would need a doc id also

Q6:6: GET /:user/documents

This is safe. Get requests are always safe as they do not change the current state. no matter how many times you call it. This also defines the idea of safety in HTTP methods. = no modifications allowed

~~POST /:user/documents~~

PATCH /:user/documents/:doc Not always

This method is idempotent. Once the request is sent, there may be modifications made, but if you send it numerous times after, it will give you the same response. Idempotency is when you send the same request and get the same answer.

4 Nice example

4