CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT342	Web Development			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
286		24/10/2022	8	5

* 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	18	
2	X	21	
3	X	14	
4	X	10	
5	X	10	
6	X	9	
7	X	8	
8	X	10	
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		100	

1

line 12:
app.patch('/wiki/pages/:page', function(req, res){ -2

lines 13-17:
var update = req.body;
var index = 0;
for (var i=0; i<pages.length; i++){
~~if (pages[i].name ==~~
if(pages.title == req.params.page)
index = i;
if(update.content){pages[i].content = update.content;}
if(update.attachments){pages[i].attachments = update.attachments;}
}
}
res.status(200).json(pages[index]);

line 20:
app.post('/wiki/pages/:page/attachments', function(req, res){ -2

lines 21-25
var attachment = req.body.attachments;
var index = 0;

for (var i=0; i<pages.length; i++){

if(pages[i].title == req.params.page){
index = i;
pages[i].attachments.push(attachment);
}
}

res.status(201).json({'attachments': pages[index].attachments});

1

2.1

With POST /wiki/page I would expect to be able to create a new page and with the data provided in the request and receive a response with status code 201 in a succesful scenario. With POST /wiki/pages/:page/attachments I would expect to create an attachment for the specified page and with the data provided in the request body, and then, to get a response with status code 201 in a succesful scenario

4

2.2

~~The GET method is should be safe, so it should ~~, so it should not modify any resource. Also~~,
Since the GET method should be safe, the browser might cache its result and the counter might not get updated and in a REST style the GET method shouldn't modify resources since it should be safe. A better way of implementing this, would be to use the POST method instead, to create a new visit to that wiki and return the length of the list of visits that the wiki has in the response. 4

2.3

A RESTful application must only allow the user to change or access resources via representations. Allowing users to pass MongoDB queries violates this constraint. Also violates the constraint that RESTful applications must be resource oriented. ~~It is~~ And it can be quite dangerous to allow this since a user can delete entire collections for example.

6

2.4

Safe methods do not result in resource modification. Idempotent methods allow resource modification and if identical idempotent methods are executed, the outcome will be the same as if it was only executed once.

GET /wiki/pages/:page Should be the only safe method out of the five and it should also be idempotent

PUT and DELETE /wiki/pages/:page Should be idempotent but not safe

An additional method that should be idempotent and not safe is:

DELETE /wiki/pages/:page/attachments

7

15 - orange
16 - black
17 - yellow
18 - black
20 - green
22 - purple
23 - pink

1

```
7   newTitle                                                    1
8   renamePage()                                                1

26  axios.put(`/wiki/page/${title}`, {'newTitle': this.newTitle, 'content': this.content})   2 2 2
27  this.title = this.newTitle                                  2
28  this.newTitle = ""
```

1

User enters URL in the browser:
GET / HTTP/1.1
Host: http://dit342-exam.se
Accept: text/html

HTTP/1.1 200 ok
Content-type: text/html

<!DOCTYPE>
<html> ... </html>

User clicks the button: method?
GET /api/camels HTTP/1.1
Host: http://dit342-exam.se
Accept: application/json

HTTP/1.1 200 ok
content-type: application/json

{'camels': [{'position':1, 'color': 'blue'}, ...]

1

The idea ~~behing~~ behind RWD ~~is~~ is to create a single front-end that adapts to the end user device instead of maintaining a front-end for ~~a~~ each device. Adapting to ~~a~~ device ~~~~ cuts the cost of multiple front-ends and include all devices.

~~To example of how~~
Two examples of how RWD principles can be used to implement accessibility are hidding purely aesthetic html elements with media queries so that screen readers can skip them and the other one is ~~the~~ to add an alternative style that increase the contrasts to increase readability.

Two examples of web accessibility implementation, that are independent of RWD, principles are providing alternative ~~text~~ description to all images and ~~or~~ avoiding time sensitive information, so users that need more time to read or to ~~reakt~~ react to the displayed data ~~be~~ can have a good experience.

7) Before Deployment automated unit tests could be made to test small parts of the new feature. If all of them pass ~~, the~~ integration tests can be run to check if the new feature in the API doesn't break ~~the code~~ the ~~front~~ API interaction ~~to the~~ with the database.

After deployment a canary ~~test~~ release can be done to check how does the system behaves but only affecting a smaller percentage of the users ~~so~~ by deploying the system with the new feature to only this fraction of users. Or, it can be deployed but use bots to simulate the traffic of users in the system in the system with the new feature to test it without affecting real users.

TCP establishes and maintains a duplex connection between 2 points. It ensures that the data delivery happens in the correct order and without packet loss (or reduced). This generates an overhead. TCP is used ~~for~~ by HTTP, since it is important that the transmitted data in a post or update request, for example, is not lost.

UDP, in the other hand, has one sender and multiple listeners, which may respond, but the session isn't maintained. UDP does not avoid loss of data as much as TCP neither the order of this data, making each transmission cheaper compared to TCP. This enables more data sent in a smaller interval. UDP is used by streaming applications, since they need to transmit a lots of data really fast and losing some pot packets is not a source of significant problems for the system