CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT 342	Web Development			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DIT342-0008-UXU		28.10.2024	12	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	15,5	
3	X	9	
4	X	18	
5	X	9	
6	X	8	
7	X	6	
8	X	10	
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		93,5	

[redacted]

1

Task 1

1.1)

Line 16:

app.patch("/milestones/:id", function (req, res) {

Lines 17-23:

const milestoneId = req.params.id

const data = req.body

const milestoneIdx = milestones.findIndex(milestone =>
milestone.id === milestoneId)

if (data.name) {
milestones[milestoneIdx].name = data.name
}

if(data.deadline) {
milestones[milestoneIdx].deadline = data.deadline
}

if(data.tasks) {
milestones[milestoneIdx].tasks = data.tasks
}

return res.json(milestones[milestoneIdx])

1

1.2)

Line 26:

app.post("/milestones/:id/tasks", function(req, res){

Lines 27-33:

const milestoneId = req.params.id

const newTask = req.body.tasks

8

const milestoneIdx = milestones.findIndex(milestone =>
milestone.id === milestoneId)

~~foo~~

milestones[milestoneIdx].tasks.push(newTask)

return res.status(201).json({tasks: milestones[milestoneIdx]
.tasks})

1

Task 2

2.1)

POST /milestones

I would expect to use this API method to create a new milestone. I expect to be able to pass new milestone data in the request body.

On success I would expect to get a 201 status code with just created resource. On failure I would expect 400 when an incorrect data was passed in the request body. (4)

POST /milestones/:milestone/tasks

I would expect to use this API method to create a new task resource that will be associated/related to an already existing milestone.

I would expect 201 status code and just created resource in the response on success. On failure I would expect either 400 (incorrect data) or 404 when milestone with the specified Id could not be found.

1

2.2)

GET /milestones/67f60a2/tasks/001

A 404 response from this endpoint could mean that either a task with id „001" does not exist or that a milestone with id „67f60a2" does not exist.

GET /milestones/67f60a2/tasks

A 404 response in this case means that a milestone with id „67f60a2" does not exist.

(3)

In this scenario I would expect the endpoint to return an empty array when there are no tasks associated with an existing milestone. Rather than returning a 404 (since we are not trying to access a specific task resource)

2.3)

GET /milestones/d828fd8/tasks?status=completed (3)

2.4)

2xx - success response - returned when everything went as expected

3xx - redirect - returned when for example a resource was moved (4)

4xx - client error - returned when a client made an error by providing wrong data, not being authorized, authenticated, etc.

5xx - server error - returned when a problem occurred on the server side. Client cannot do much about it.

2.5)

The „main" ones are: GET (0,5), PUT (0,5), DELETE (0,5) (1,5)

CHALMERS	Anonymous code Anonym kod DIT342-0008-UXU	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylles av lärare) 9	Consecutive page no. Löpande sid nr 5 Question no. Uppgift nr 3

Task 3

Line 14: blue

Line 15: purple

Line 16: ~~aaaa~~ orange

Line 17: purple

Line 18: green

Line 19: purple

Line 20: purple

Line 22: pink

Line 25: black

1

Task 4

Line 9:

```
<input type="text" v-model="task" />
```
1

Line 10:

```
<input type="text" v-model="due" />
```
1

Line 11:

```
<button @click="addTask">Add task</button>
```
1

Line 13:

```
<p><ul><li v-for="task in tasks">
```

Lines 14 - 15:

```
<p :class="overdue(task.due) ? 'overdue' : ''">
  {{ task.task }} {{ task.due }}
<</p>
<button type="button" @click="completeTask(task)">
  Done!
</button>
```
6

Line 28:

```
this.tasks.push({
  task: this.task,
  due: this.due
})
this.task = ""
this.due = ""
```
3

1

Line 31:

```
this.tasles = this.tasles.filter(t => t.task !== task.task)
```

3

Line 34:

~~return due > this.overdueDate~~

```
return Date.parse(due) > this.overdueDate
```

3

returning bool matches usage

Task 5

Large browser window - 1300px

Div 1	Div 2	Div 3	Div 3	Div 3							
Div 4	Div 5	Div 5	Div 5	Div 5	Div 5	Div 5	Div 5	Div 5	Div 5	Div 6	Div 6

5

Small browser window - 530px

Div 1	Div 2	Div 2	Div 2	Div 2	Div 2	Div 2	Div 3	Div 3	Div 3	Div 3	Div 3
Div 4	Div 4	Div 4	Div 4	Div 5	Div 5	Div 5	Div 5	Div 6	Div 6	Div 6	Div 6

4

1

Task 6

An IP stands for Internet protocol and is the most important protocol on the 3 Layer of the OSI model (the network layer)

In general, IP is responsible for routing and addressing of data packets between nodes/clients over networks. It is a connectionless protocol (one payload is sent at a time) and an unreliable protocol (data may or may not get to the other client)

TCP stands for Transmission Control Protocol and is a protocol mainly working on a 4 layer of the OSI model (the transportation layer).

The main job of TCP is to deliver packets between two nodes (it is a unicast protocol). TCP is reliable (makes sure that packets are delivered to the receiver, lost packets are re-sent) and ordered (packets reach their destination in the same order they were sent)

1

Task 7

Cookies are key-value pairs set by the server and then included in each client's response. 1

First page request (1)

GET / HTTP/1.1 ← client calls an endpoint
Host: localhost:3000
~~Accept: application/json~~ 2

Response with a cookie (2)

HTTP/1.1 200 OK ← server responds with 200 OK and sets a cookie that is included in the response header 1
Set-Cookie: foo=bar

Request to the server after a cookie was set (3)

GET /users HTTP/1.1 ← client stored/saved the received cookie from the server in the cookie storage and will include it ~~with~~ in the header with the next request 1
Host: localhost:3000
Cookie: foo=bar 1

valid domain, expiry

1

Task 8

On-premise hosting means that, for example a company buys required hardware and setup their own servers and infrastructure that they could use for their own needs.

Cloud hosting on the other hand allows the company to „buy" those servers, tools, and initialy setup infrastructure from one of the cloud providers.

Key differences between those approaches

On-premise hosting:

-> big (or biggest) up-front costs related to buying required hardware and setting up an infrastructure

-> on-premise hosting servers needs to be maintained and updated

-> a company has to have an actual space for servers in their office

-> high flexibility and independance about how those hosting resources can be utilized

-> potentially smaller costs in the long run in comparison to cloud hosting

Cloud hosting

-> small up-front costs (pay only for what you use)

-> no need to update and maintain the servers and the core infrastructure

-> no need to have space for actual servers

-> less flexible

-> prone to vendor-locking and being dependent on the cloud provider's environment

-> potential sky-high costs in a for long run related to computations, capacity, and load.

-> faster and easier to start working with especially when a cloud provides provides IaC which can be used to setup an infrastructure and control it using code, rather than doing it manually.

In general, on-premise hosting is more expensive up-front and requires more work to run. However, it provides total flexibility and independence for the company. Cloud hosting is cheaper up-front and already comes with an initial / core infrastructure which make it easier to start working with. However, is less flexible and might close the company in the provider's tech ecosystem.