CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT342	Web and mobile development			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DIT342-0011-LUG		14/8-23	8	4

* 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	26	
2	X	~~20~~ 17	
3	X	12	
4	X	~~8~~ 9	
5	X	6	
6	X	11	
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points		25 + BONUS	

[redacted]

1. This implemetation violate the principle of uniform of interface that an API desigen should adhere to to be concidered Restful. Since this implemention Vaiolate the sepration of concern as well as safty and idempotensy. The post request is unsafe and not idempotent. which means that it will be concidered state changer for every request send and hence cloud introudce unexpected behavior if it was tested by an automated test tool for instane. On the other hand it should be implemented with respect to the method names and behavior. where get is only responseble for getting resourses from the server, post only when the client want to create a new resource, put when updating an entier resourse is desierd, Patch when partielly update is desiered, and finaly delete when deleting a resource is wanted. By doing so the application whould be ~~functionally~~ Function Correctlly and produce expectibale outcoms that are easy to intract with and implement proper error handing due to thier predicteble behaviors as was recommnded by Roy Fieldent famous PHd dissertion and API design convenstions.

~~This~~

The implmetion provided dose not satisfy the afromentiod aspects, hence it's problematic.

1

2.
```
app.post('/wiki/pages', function(req,res) {
    const newPage = {
        title: req.body.title,
        content: req.body.content
    }
    const pages.push(newPage)
    res.status(201).json({id: pages.length-1})
})

app.get('/wiki/pages', (req,res) => {
    return res.json(pages)
})

app.update('/wiki/pages', (req, res) => {
    var body = req.body
    var id = req.query.id
    pages[id] = body
    res.json({id: id})
})
```

13

3.
```
app.delete('/wiki/pages/:pageId', (req, res) => {
    const pageID= req.params.pageId;
    const updatedPages= pages.filter(p => p.id !== pageID)
    res.json(updatedPages);
})
```

pages = uppdated pages.

8

26/30

1

2.

- line 3 h1 ✓
- line 4 div
- line 5 .button ✓
  ~~&~~ media query
- lines 27-30 #listing

```
<div>
<label ~~name="button1"~~ id="button1" value="button1" class="button">
FirstButton:</lable>

<button name="button1" ~~type="submit"~~ onClick="buttonPressed(1)">

Button 1 </button>
</div>
<div>
<label name="button2" id="button2" class=button">Second Button:
</label>
<button name="button2" type "~~submit~~" onClick="buttonPressed(2)">

Button 2</button>
</div>
```

77
~~100~~ / 22

1

- lines 10-12

```
<tr v-for(page in pages) > <td><b> {{ page.title }} </b></td>
<td><b> {{ page.content }} </b></td> </tr>
```

✓ 4

- lines 14-16

```
<form submit="onSubmit" >
<input name="Title" v-model=title type="text"> EnterTitle</input>
<input type="text" name="content" v-model = content> Enter content</input>
<input type="submit @="addNewPage" > Add New Page</input>
</form>
```

- lines 28-30

```
metods: {
    addNewpage() {
        const newPage = {
            title: this.title,
            content: this.content
        }
        this.pages.push(newPage)
    },
created: {
    const pages = []
    for(let i=0; i< this.pages.length; i++) {
}
```

✓ 4 ~~4~~

12/12

4.

A user send a get request to the DNS www.my-blog-space.se where usually the user expect an HTML page to be loaded In the request the ~~user~~ request indicate what data format the user accept upon a get request such as this one.

The server sends back a responce with status code 200 "Ok" if the domin name was valid and load the login page as was asked in the get request, alternitvly a status code of 404 if the domin name was wrong or not avalible for instance.

upon success senario the user send a post request containing the user username and password, assuming that was been asked in the login page, in the request body.

If the user credintial was valid the server sends back a responce with status code 200 "Ok", and allow the the user to access itsown blag page.

Otherwise the server sends back a responce with Statu code 403 "unauthrized" and proper error message indicting that the user did not provid correct credintials.

The requests and responses occur throug HTTP protocol, ~~and~~. For authenticution many libraries cloud be used, ie. jwt. + library calls

5

4.

Request:

Get ~~http://www.my-blog-space.se~~ HTTP/1.1

Host: 'www.my.blog-space.se'

Accept: text/HTML, */>

Responce:

HTTP/1.1 200 OK

Host: 'www.my-blog-space.se'

Content-typ: text/HTML, */>.

* Body

4

9

~~8~~ / 14

5.

1. Include a description for all images in the alt="" attribute of <img> tags to help visually impared indevidules to get access to the images content throug speaking tools in the browser for instance. or users with bad intenet connection. to undestand the images content.
2. Add media querys to make the website ~~more~~ accessble for all screen sizes so the users with small screen size cloud have better user experience and remove unnessary cantent that cloud worsen the user experien on small screen sizes or ~~des~~ desterb importent functionality
3. ~~Add~~ tabs so the only importent ones upload first ~~if the~~ its for visually impared users.

t more

6/8

6.

layer 1 ~~phys~~ physical layer

contains the actual lines and phsyical components that the communication pass through.

Layer 2: Data-link layer

~~make~~ includs the infrasture that connect the component in the physical layer.

Layer ~~3~~: Transmission layer

includes the infrastructure that connect mutible hops, Technologies QoS "quality of service" that ensure that messagen cloud be delevired if the connection was lost for insanste

Layer 3, Network layer:

connect multible node on the network enabling connection between clients and servers through intenet protocol IP for instance.

Layer 5:

Session

11/14

Layer 6: presentian layer

where clients and servers ~~cant~~ communicate through diffrent internet protocols that are on top of ~~HTTP~~ IP such as TCb, UDP, and HTTP. The communtion occur through requests and responces in the case of HTTP~~s~~ for instance.

Layer 7: application layer

where Api ~~that~~ take place to orgnize ~~of~~ the connection between client and servers that happens upon requests and ~~respo~~ responses. using HTTP protocal for instance.