

The preview shows how the test is presented to participants. You can also see the correct answers for all questions except free text answers. No correct answers are shown to participants when they complete the test.

Total max score: 30

Start by reading through all questions. Peter will visit the room 45 minutes after the exam has started to clarify the questions you do not understand. Max score is 30 points.

- For grade 3, 40% of max score (12 points) is required.
- For grade 4, 60% of max score (18 points) is required.
- For grade 5, 80% of max score (24 points) is required.

You are not allowed to use the computer for anything else but answering the questions on this page.

Write your answers in either English or Swedish. If you write your answers in Swedish, make sure to not introduce any translation confusement. Write proper sentences (spelling, upper/lower case characters, punctuation, etc.). Answers that do not do this good enough/are vague/are ununderstandable cannot receive full score on the questions.

Good luck!

In client-side JavaScript, one often use the event `DOMContentLoaded`. Explain why web developers often use this event, and suggest a workaround for these developers if they don't want to use this event.

Max score: 2

In client-side JavaScript, event objects has a method called `preventDefault()`. Explain the result of calling this method, and give a practical example of when you would need to call it (no need to write any code, just describe a practical example using words).

Max score: 1

In the client-side JavaScript code below, the function `getAccountById()` sends an HTTP request to a REST API to retrieve information about an account with a specific id.

```
getAccountById(23, function(error, account){
  if(error){
    alert("Couldn't fetch the account, check your Internet connection.")
  }else if(account == null){
    alert("No account with id 23 exists.")
  }else{
    document.getElementById("username").innerHTML = account.username
  }
})
```

Explain what type of security vulnerability the code contains, and give an example of how a hacker could exploit it (no need to explain a specific attack, just explain what the hacker would do in general to exploit the vulnerability).

Max score: 1

Explain what docker images and docker containers are.

Max score: 1

Explain how a three layered architecture works. What is the name and responsibility of each layer, and what technologies do we use in each of them in web applications?

Max score: 3

In a web application built on a three-layered architecture, in which layer(s) do we use the MVC pattern? Justify your answer.

Max score: 1

In a three-layered architecture for a web application, in which layer(s) would we write code protecting us against XSS attacks? Justify your answer.

Max score: 1

A data access layer communicating with a MySQL database exposes a function called `selectAllFromTheAccountsTable()` another layer can use to retrieve all posts from the `accounts` table in the MySQL database. Explain why this is a poorly chosen name.

Max score: 1

Give an example of data that are better stored in a none relational database (of some type) instead of a relational database. Justify your answer.

Max score: 1

Here is some code that allows any user to (anonymously) post comments on other users' profile pages.

```
app.post("/users/:id/comments", function(request, response){

    const userId = request.params.id // The id of the user the comment is for.
    const signature = request.body.signature // The name of the one writing the
comment.
    const comment = request.body.comment // The comment itself.

    // Let us assume the user has entered a good signature
    // and comment, so we don't need to validate them.

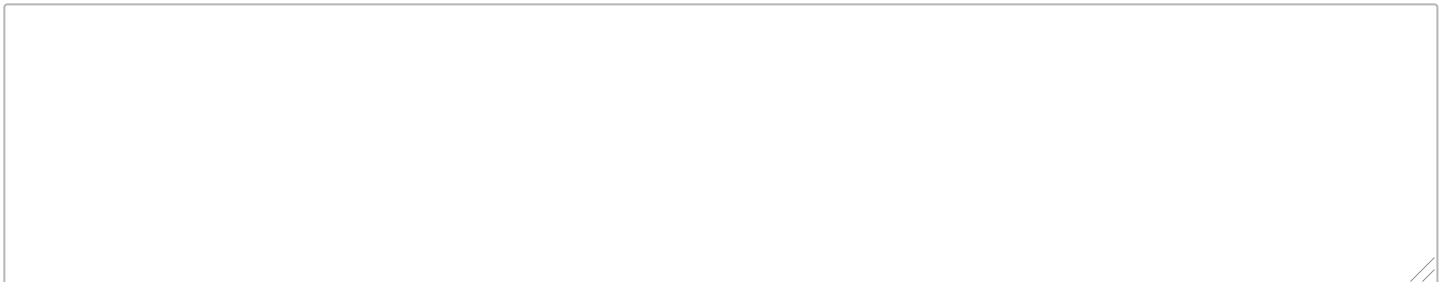
    commentRepository.createComment(userId, signature, comment, function(commentId)
{

    response.send("Created comment got id "+commentId+".")

    })

})
```

Given that the communication with the database that stores all comments work, the code above still contains a flaw. What is that? What needs to be done to fix it?



Max score: 1

A web application implementing a REST API allows you to update an existing guestbook post by sending a PUT request to `/api/guestbook` with information about the guestbook post that should be updated (id, signature and post) in the body of the request. Is this a good or bad design of the REST API? Justify your answer.

The same web application also allows you to delete a guestbook post with a specific id by sending a DELETE request to `/api/guestbook`, and in the body of the request specify the id of the guestbook post that should be deleted. Is this a good or bad design of the REST API? Justify your answer.

Max score: 2

What is the main difference between *vertical scaling* and *horizontal scaling*? Can an application that can be scaled horizontally always easily be scaled vertically as well? Justify your answer.

Max score: 2

Imagine we have the code below with the variable `queryCounter` to keep track of how many SQL queries we have sent to the database:

```
let queryCounter = 0

app.get("/accounts", function(request, response){

  const oldQueryCounter = queryCounter

  const query = "SELECT * FROM accounts"

  db.getAll(query, function(accounts){

    // Let's assume nothing is wrong with the connection to the database.

    response.render("accounts.hbs", {accounts: accounts})
```

```
queryCounter = oldQueryCounter + 1
```

```
})
```

```
})
```

```
// And similar for all other requests that involves sending queries to the  
database.
```

Even if we only run the web application on a single server, the counting does not always work as it should. Why not?

Max score: 1

OAuth 2.0 and OpenID Connect are both specifications that can be implemented in REST APIs. When should you use which one? Or put in another way: what functionality does each of them specify how you should implement in your REST API?

Max score: 1

OAuth 2.0 defines four different ways one can obtain an access token. Describe how the two ways *Authorization Code Grant* and *Implicit Grant* works, and when you should use which one.

Max score: 2

What is the difference between a self-contained token and a token that is not self-contained?

Max score: 1

Does it make sense to put a user's favourite color in an access token? Justify your answer.

Max score: 1

Does it make sense to put a user's administration privileges in an ID token? Justify your answer.

Max score: 1

Explain what a claim in a JSON Web Token is.

Max score: 1

JWT defines three different categories of claims. Which are they, and how do they differ? Which ones are most often better to use? Justify your answer.

Max score: 3

Explain what the *Same-Origin Policy* is, and how *Cross-Origin Resource Sharing* is related to it.

Max score: 2