CEN467-XML & Web Services

Project Name: Catching Santa Claus

Group Members - Student Numbers:

Nahide Sena Sabırlı - 20091000020
Elif Naz Kadayıfçı - 20091000038
Rawan ElShenieky - 21091000188

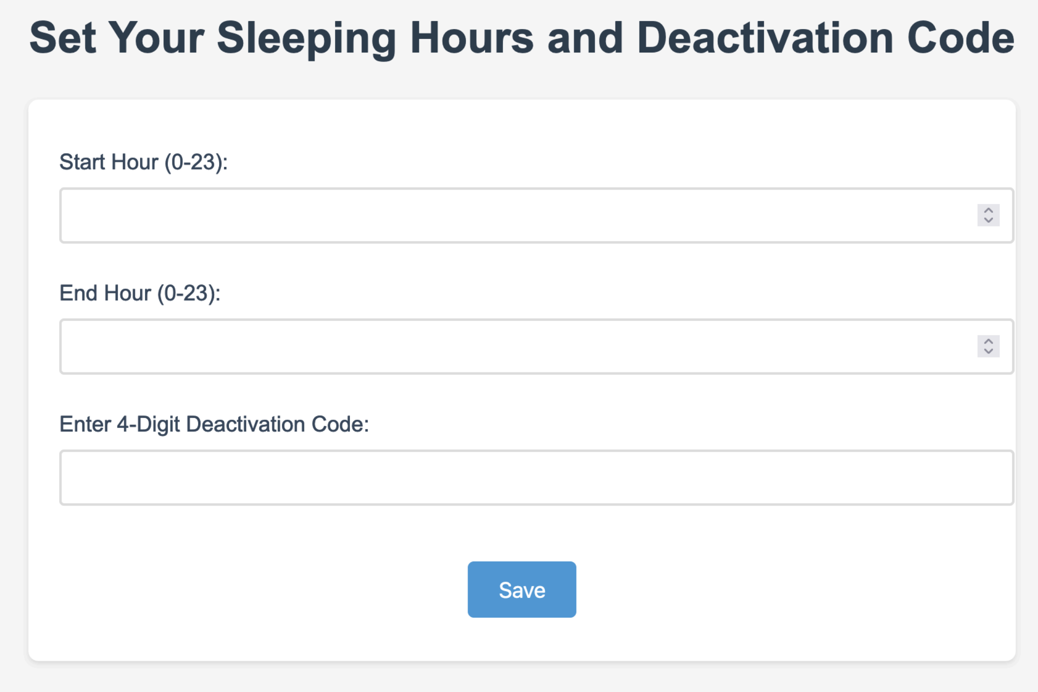
Introduction

Our project is designed to help us catch thieves (AKA Santa Claus) that come at night while we are asleep. We have four files which are: sleeping_hours.php, style.css sleeping_hours.xml, and all_logs.xml (added after comments received in the presentation). Our full setup works in sync with a Wokwi simulation and uses HTTP protocol. Having a simple, clear, and critical purpose, we present to you our system in the upcoming sections.

How does it work?

Step 1

The user first sees the page below, prompting him/her to enter their sleeping hours based on 24-hour system (no PM and AM) as well as a deactivation code of 4-digits.



Set Your Sleeping Hours and Deactivation Code

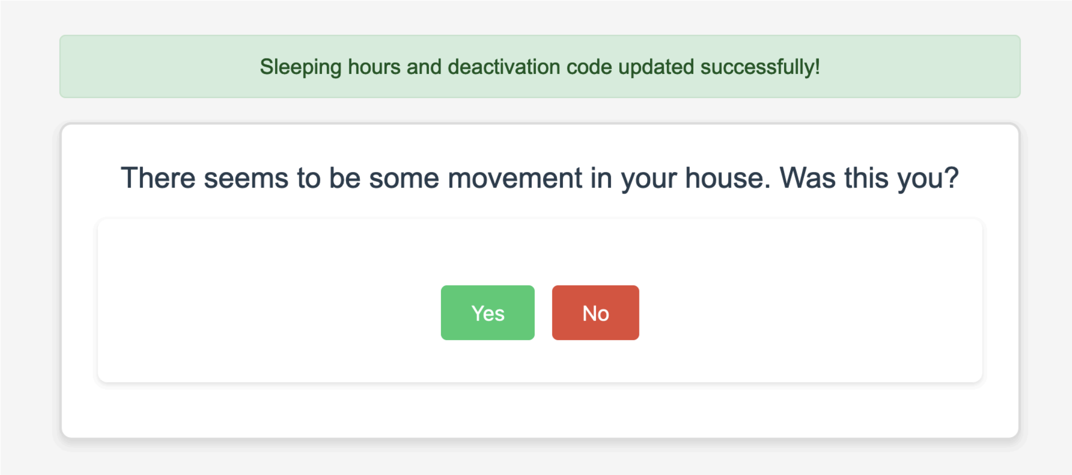
Start Hour (0-23):

End Hour (0-23):

Enter 4-Digit Deactivation Code:

Step 2

By clicking on the save button, the user's responses are saved in "sleeping_hours.xml" and "all_logs.xml". All details of "all_logs.xml" will be explained in the "Additional Touch" section of this report. Then a confirmation pops up along with a notification stating that there have been motion detected in the user's house, asking them to state if this motion was caused by them or not.



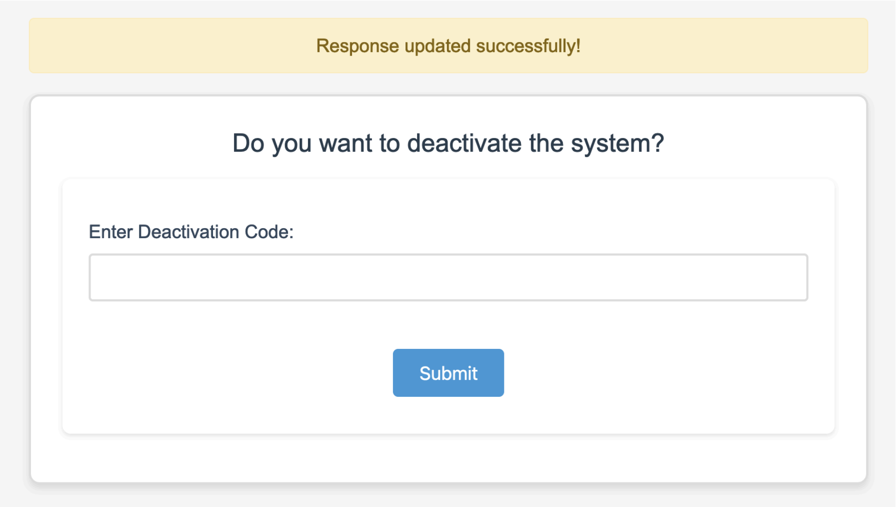
Sleeping hours and deactivation code updated successfully!

There seems to be some movement in your house. Was this you?

In case the user presses yes, the system will show a message saying that the security system is deactivated, returning them to the main first page. The xml file will be updated for the values of "response", "deactivation", and "codematch" elements, changing from "none" to "yes" for all three of them.

Step 3

Otherwise, if the user presses no, then they are directed to a page that asks them to enter the deactivation code if they want to deactivate the system.

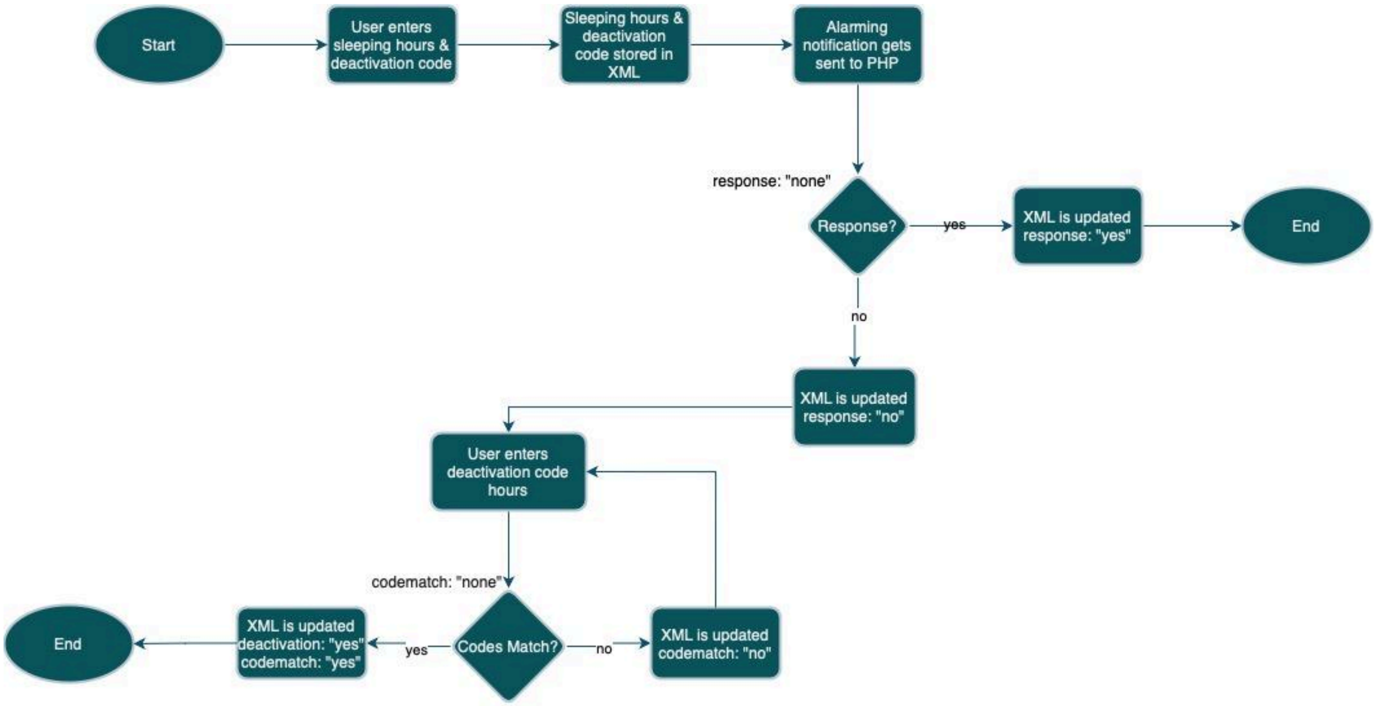


Response updated successfully!

Do you want to deactivate the system?

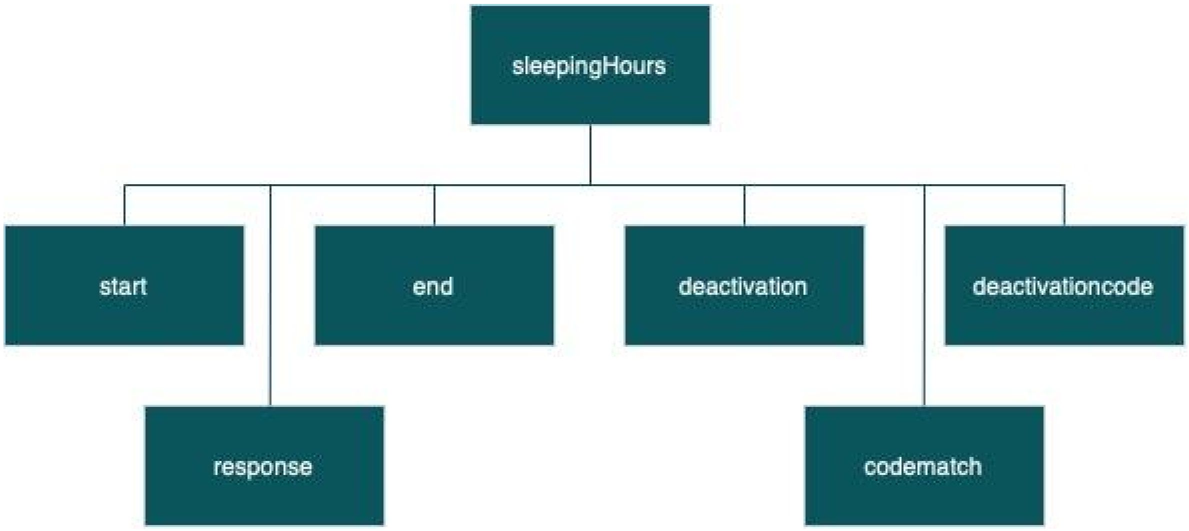
Enter Deactivation Code:

The value of “response” change from “none” to “no” in the xml file.If the deactivation code stored in the xml file, that was previously chosen by the user, does not match the one they entered at this step, the system shows a message saying that the code is incorrect and prompts the user to re-enter the code again. At this point the “codematch” element’s value goes from “none” to “no”. Once the user enters the correct deactivation code, the “codematch” element’s value goes from “none” to “yes” and the system will show a message saying that the security system is deactivated successfully, returning them to the main first page.



Start
User enters sleeping hours & deactivation code
Sleeping hours & deactivation code stored in XML
Alarming notification gets sent to PHP
response: "none"
Response?
yes
XML is updated response: "yes"
End
no
XML is updated response: "no"
User enters deactivation code hours
codematch: "none"
Codes Match?
yes
no
XML is updated deactivation: "yes" codematch: "yes"
XML is updated codematch: "no"
End

Elements Explained

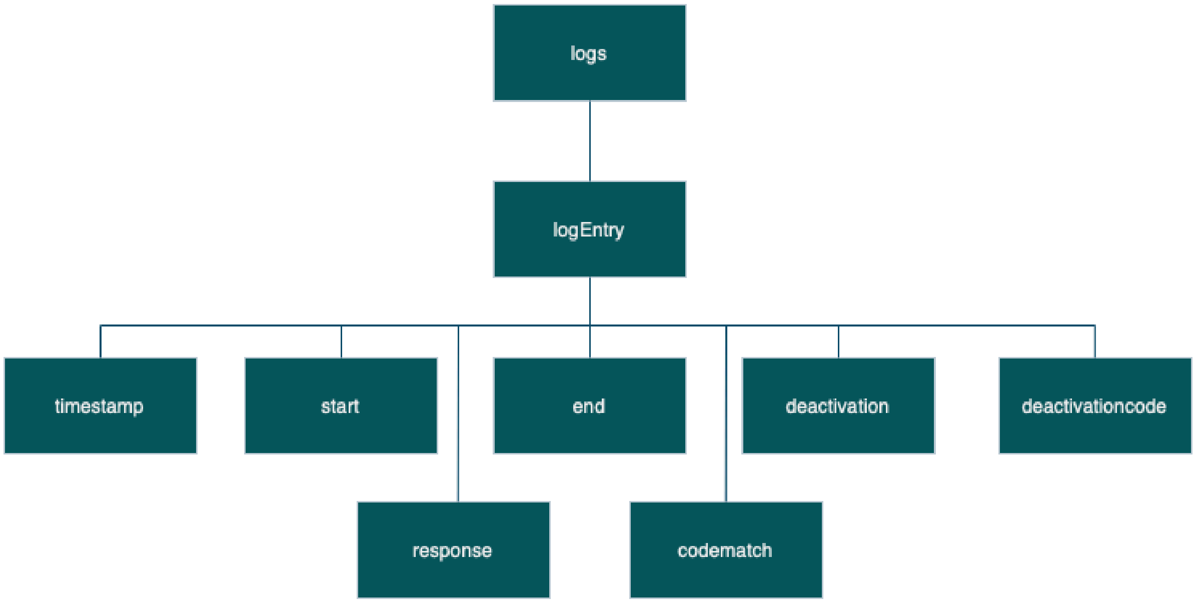


In “sleeping_hours.xml” we had our tree structure as shown below:

“start” and “end”: are important to define the sleeping hours of the users
“response”: stores the user’s response for the question “was this you?”. It has initially the value “none” while the user is still in step 1.
“deactivationcode”: set by the user in step 1. It is essential for security reasons that the user does not directly deactivate the system without the deactivation code.
“deactivation”: is changed when the security system is deactivated.
“codematch”: checks whether the codes entered by the user in step 1 and 3 are the same.

Additional Touch

Based on your suggestion to store the user’s logs in a separate xml file, we successfully implemented it and named our new xml file as “all_logs.xml” which has additional elements to the ones in “sleeping_hours.xml” which are logs (root element) logEntry (parent element), and timestamp (child element). Leaving the structure of the file to be as shown in the figure below:



Notes & Comments

Suggested scenario

While planning for our project we had a specific setup set for increasing the optimal outcome of our system. We assumed that the sleeping hours are set on a daily basis, meaning that the user can change their sleeping schedule. Also, it is designed for one user only, meaning that if person A set up their sleeping schedule and the motion got detected in A’s house and at the same time another person B enters the system, they will directly be in Step 2 with person A’s information saved.

Link with IoT project

This project is prepared to be used in the Internet of Things course with code CEN479. The reports of each subject contain differences in terms of content as each one focuses more on its relevant course core.

Alternative Scenarios

Our system can be implemented in different environments other than in home while sleeping. For example, it can also be implemented in companies, offices, malls, etc… The hours that will be set can be working hours or break times.

Hosting Details

Based upon your suggestion of using infinityfree.com, we used it to host our webpage.
You can check it out from the following links:
The main webpage: http://santaclaus.lovestoblog.com/sleeping_hours.php
The sleeping_hours.xml: http://santaclaus.lovestoblog.com/sleeping_hours.xml
The all_logs.xml: http://santaclaus.lovestoblog.com/all_logs.xml
The Wokwi project: https://wokwi.com/projects/418091855523480577
Note that the links are accessed only through the computer and it does not work on the phone.

Possible Improvements

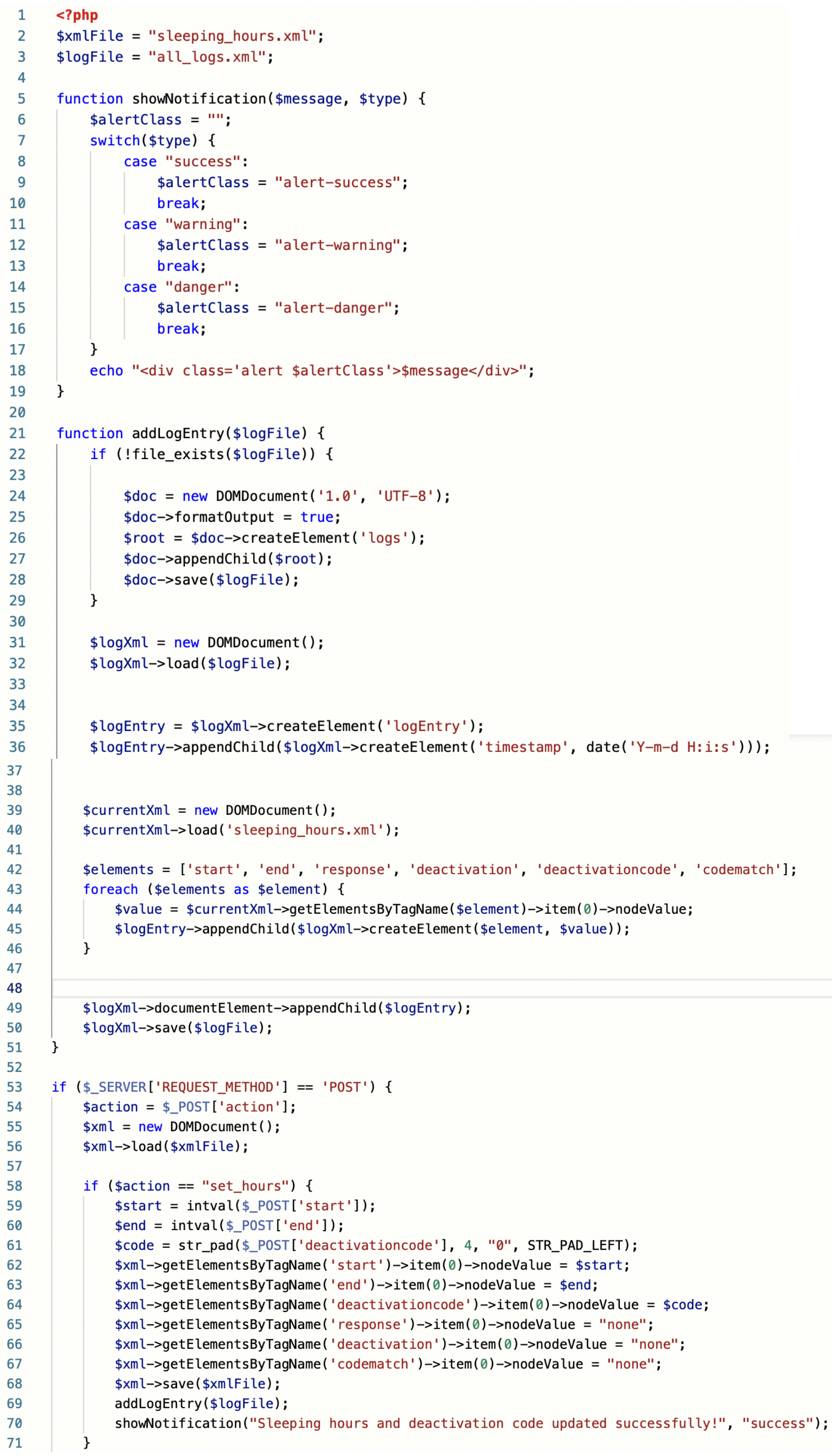
In terms of security, we can upgrade our webpage so that it has user special credentials, which are generated from the system itself based on the user’s ip address. We can also add the duration and the specific time in which the system had been activated, how many times there were wrong attempts for re-entering the deactivation code in step 3, and including different types of data like photos captured during the activation time of the system. There can be a live-streaming video that is later saved when pressing no in step 2. However not all of these upgrades can solely be made using xml.

In terms of user experience, the webpage can be achieved through phone, or smartwatch to ensure that the user can be notified through multiple devices. The user interface and the design can be enhanced using templates like the ones in w3schools or bootstrap.

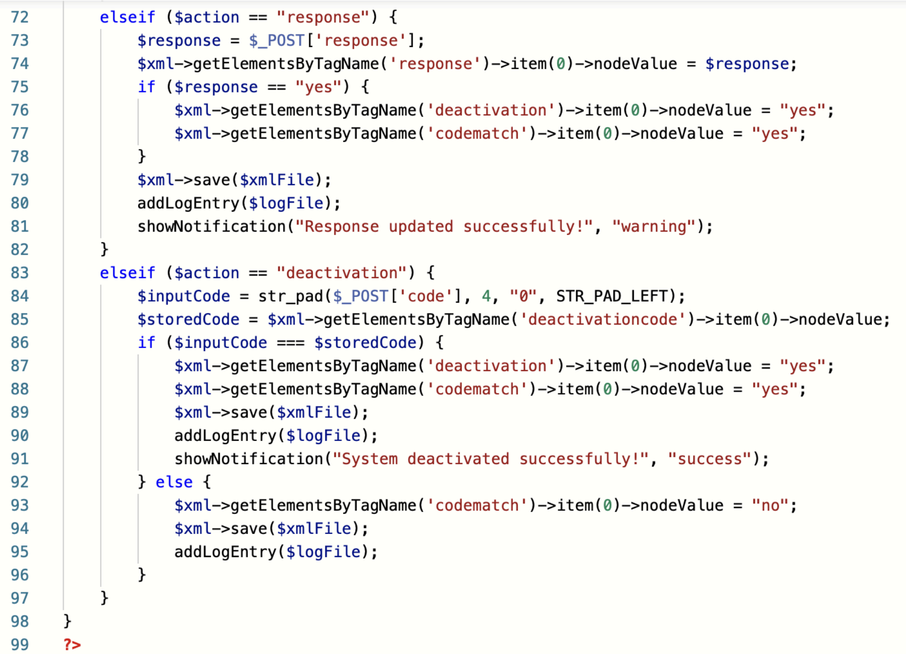
Conclusion

Creating a simple alarming security system to catch Santa Claus using XML can be just a start for many other project ideas that can be developed using XML or another language. Combining our project with IoT using Wokwi provided a simulated environment of the actual possible usage of the system, giving us a learning experience in means of enhancing the security measures and the reality of the system.

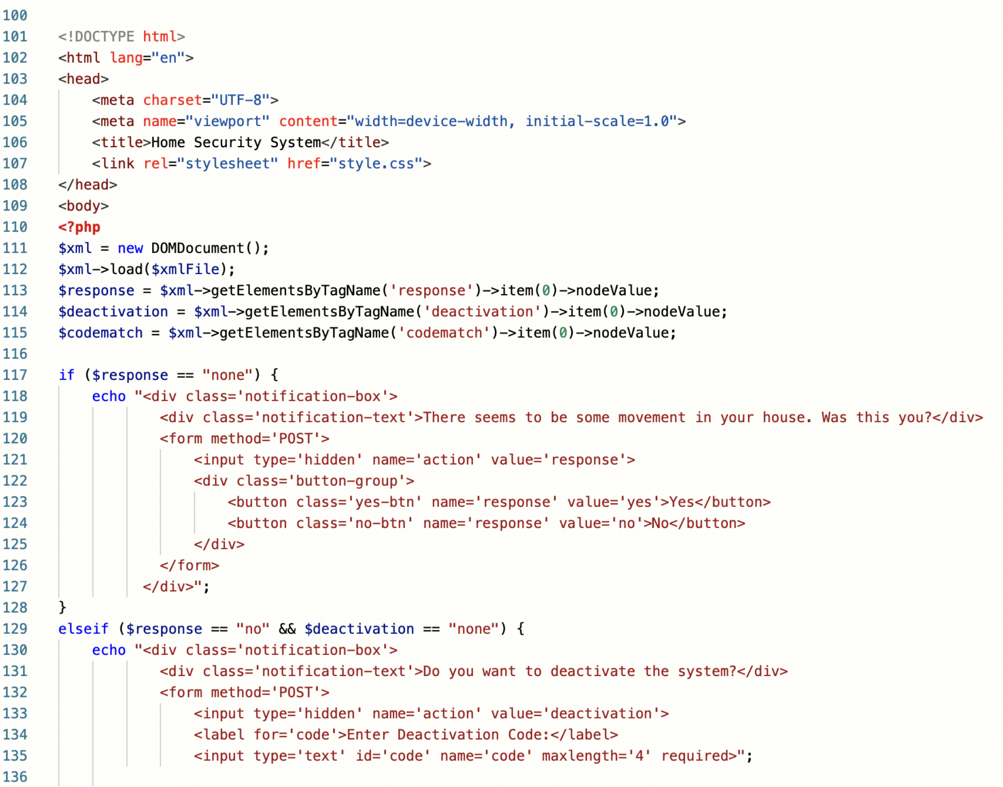
Code Snippets



```php
<?php
$xmlFile = "sleeping_hours.xml";
$logFile = "all_logs.xml";

function showNotification($message, $type) {
    $alertClass = "";
    switch($type) {
        case "success":
            $alertClass = "alert-success";
            break;
        case "warning":
            $alertClass = "alert-warning";
            break;
        case "danger":
            $alertClass = "alert-danger";
            break;
    }
    echo "<div class='alert $alertClass'>$message</div>";
}

function addLogEntry($logFile) {
    if (!file_exists($logFile)) {

        $doc = new DOMDocument('1.0', 'UTF-8');
        $doc->formatOutput = true;
        $root = $doc->createElement('logs');
        $doc->appendChild($root);
        $doc->save($logFile);
    }

    $logXml = new DOMDocument();
    $logXml->load($logFile);


    $logEntry = $logXml->createElement('logEntry');
    $logEntry->appendChild($logXml->createElement('timestamp', date('Y-m-d H:i:s')));


    $currentXml = new DOMDocument();
    $currentXml->load('sleeping_hours.xml');

    $elements = ['start', 'end', 'response', 'deactivation', 'deactivationcode', 'codematch'];
    foreach ($elements as $element) {
        $value = $currentXml->getElementsByTagName($element)->item(0)->nodeValue;
        $logEntry->appendChild($logXml->createElement($element, $value));
    }


    $logXml->documentElement->appendChild($logEntry);
    $logXml->save($logFile);
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $action = $_POST['action'];
    $xml = new DOMDocument();
    $xml->load($xmlFile);

    if ($action == "set_hours") {
        $start = intval($_POST['start']);
        $end = intval($_POST['end']);
        $code = str_pad($_POST['deactivationcode'], 4, "0", STR_PAD_LEFT);
        $xml->getElementsByTagName('start')->item(0)->nodeValue = $start;
        $xml->getElementsByTagName('end')->item(0)->nodeValue = $end;
        $xml->getElementsByTagName('deactivationcode')->item(0)->nodeValue = $code;
        $xml->getElementsByTagName('response')->item(0)->nodeValue = "none";
        $xml->getElementsByTagName('deactivation')->item(0)->nodeValue = "none";
        $xml->getElementsByTagName('codematch')->item(0)->nodeValue = "none";
        $xml->save($xmlFile);
        addLogEntry($logFile);
        showNotification("Sleeping hours and deactivation code updated successfully!", "success");
    }
```



```
    elseif ($action == "response") {
        $response = $_POST['response'];
        $xml->getElementsByTagName('response')->item(0)->nodeValue = $response;
        if ($response == "yes") {
            $xml->getElementsByTagName('deactivation')->item(0)->nodeValue = "yes";
            $xml->getElementsByTagName('codematch')->item(0)->nodeValue = "yes";
        }
        $xml->save($xmlFile);
        addLogEntry($logFile);
        showNotification("Response updated successfully!", "warning");
    }
    elseif ($action == "deactivation") {
        $inputCode = str_pad($_POST['code'], 4, "0", STR_PAD_LEFT);
        $storedCode = $xml->getElementsByTagName('deactivationcode')->item(0)->nodeValue;
        if ($inputCode === $storedCode) {
            $xml->getElementsByTagName('deactivation')->item(0)->nodeValue = "yes";
            $xml->getElementsByTagName('codematch')->item(0)->nodeValue = "yes";
            $xml->save($xmlFile);
            addLogEntry($logFile);
            showNotification("System deactivated successfully!", "success");
        } else {
            $xml->getElementsByTagName('codematch')->item(0)->nodeValue = "no";
            $xml->save($xmlFile);
            addLogEntry($logFile);
        }
    }
}
?>
```



```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home Security System</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<?php
$xml = new DOMDocument();
$xml->load($xmlFile);
$response = $xml->getElementsByTagName('response')->item(0)->nodeValue;
$deactivation = $xml->getElementsByTagName('deactivation')->item(0)->nodeValue;
$codematch = $xml->getElementsByTagName('codematch')->item(0)->nodeValue;

if ($response == "none") {
    echo "<div class='notification-box'>
            <div class='notification-text'>There seems to be some movement in your house. Was this you?</div>
            <form method='POST'>
                <input type='hidden' name='action' value='response'>
                <div class='button-group'>
                    <button class='yes-btn' name='response' value='yes'>Yes</button>
                    <button class='no-btn' name='response' value='no'>No</button>
                </div>
            </form>
          </div>";
}
elseif ($response == "no" && $deactivation == "none") {
    echo "<div class='notification-box'>
            <div class='notification-text'>Do you want to deactivate the system?</div>
            <form method='POST'>
                <input type='hidden' name='action' value='deactivation'>
                <label for='code'>Enter Deactivation Code:</label>
                <input type='text' id='code' name='code' maxlength='4' required>";

    if ($codematch == "no") {
        showNotification("Wrong deactivation code, please re-enter", "danger");
    }

    echo "<div class='button-group'>
            <button type='submit'>Submit</button>
          </div>
          </form>
          </div>";
}
else {
    echo "<h1>Set Your Sleeping Hours and Deactivation Code</h1>
          <form method='POST'>
            <input type='hidden' name='action' value='set_hours'>
            <label for='start'>Start Hour (0-23):</label>
            <input type='number' id='start' name='start' min='0' max='23' required>

            <label for='end'>End Hour (0-23):</label>
            <input type='number' id='end' name='end' min='0' max='23' required>

            <label for='deactivationcode'>Enter 4-Digit Deactivation Code:</label>
            <input type='text' id='deactivationcode' name='deactivationcode' maxlength='4' required>

            <div class='button-group'>
                <button type='submit'>Save</button>
            </div>
          </form>";
}
?>
</body>
</html>
```