Coursework Semester 2: Documentation

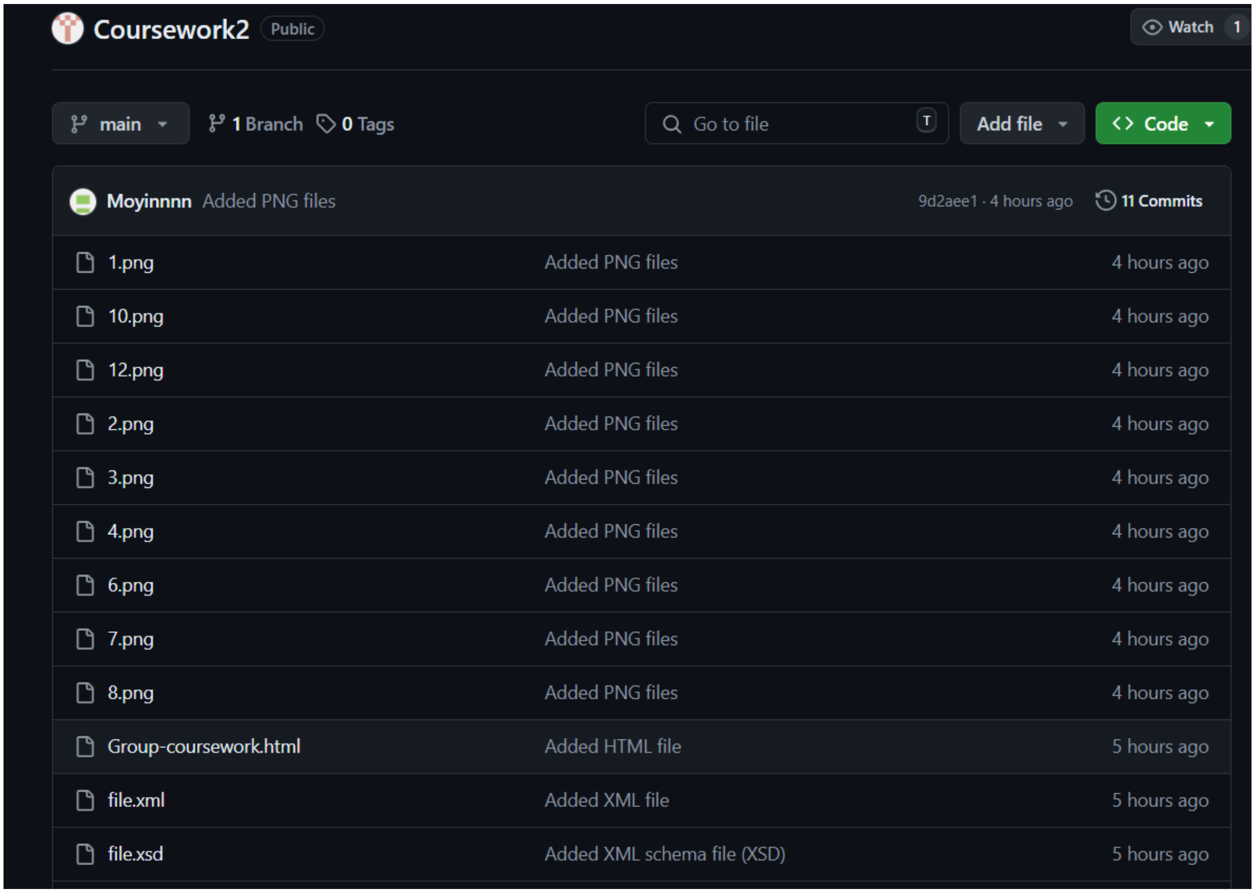
Introduction

The coursework semester assessment is a compilation of significant topics in Computer Science. In particular, the brief emphasises teamwork to execute code in HTML, CSS, JavaScript, LaTex, XML and git on GitHub. The first task, git demands the use of a version control repository to collaboratively work on the different areas of the brief then a changelog file is to be created to display the different commits made to the repository. Secondly, the task on HTML, CSS and JavaScript involves the set-up of a responsive and data-receptive webpage.

Furthermore, using LaTex (in our case on OverLeaf), we are expected to create a document summarising the purpose and results of the assessment given, showing our comprehension of the different topics. Finally with XML, a detailed representation of work done on the assessment is expected with utmost accuracy and organisation. Essentially, this coursework aims to build teamwork abilities whilst also gaining greater insights on some of the most important aspects of Computer Science.

Git

In this subtask, we created a public repository on GitHub ("Coursework2") where we added, named and committed the files used in the completion of the exercise. Afterwards, the log values of each file were used added to the change.log file to see the different changes that were made throughout the execution of the assessment. Initially, a branch (separate from the main branch) was created with a few files committed to it. However, these branches was later merged with the main using a pull request on GitHub.



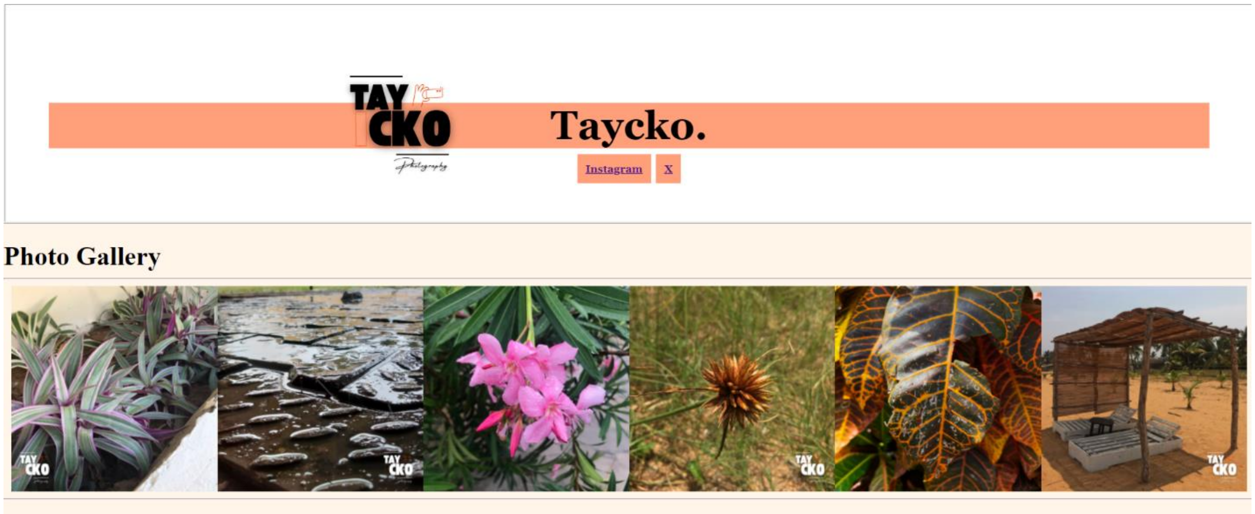
Moyinnnn Added PNG files 9d2aee1 · 4 hours ago 11 Commits

1.png	Added PNG files	4 hours ago
10.png	Added PNG files	4 hours ago
12.png	Added PNG files	4 hours ago
2.png	Added PNG files	4 hours ago
3.png	Added PNG files	4 hours ago
4.png	Added PNG files	4 hours ago
6.png	Added PNG files	4 hours ago
7.png	Added PNG files	4 hours ago
8.png	Added PNG files	4 hours ago
Group-coursework.html	Added HTML file	5 hours ago
file.xml	Added XML file	5 hours ago
file.xsd	Added XML schema file (XSD)	5 hours ago

Figure 1: GitHub repository - Coursework2

HTML, CSS, JS

To complete this task, there was the thorough implementation of HTML, CSS and JavaScript. Since there were no restrictions on the contents of the webpage, we decided to go for a photo gallery that displays the images of a photography company called Taycko. The webpage mainly consists of images, including a signup form and an “About Us” at the end of the page. Other features included links to the photographer’s social media and a logo at the top left hand of the page.

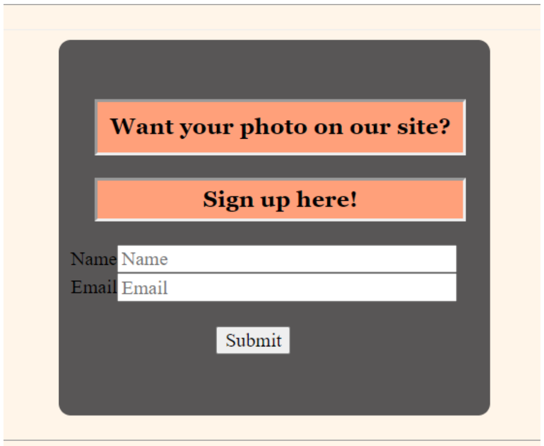


Taycko.

Photo Gallery

Figure 2: Start of webpage featuring company logo and images

Intentionally, we opted go for a minimalistic style for the website using a few but very vibrant colours. This was achieved using a diverse range of CSS elements and properties. As demanded by the brief, CSS was applied in the creation of the sign-up form.

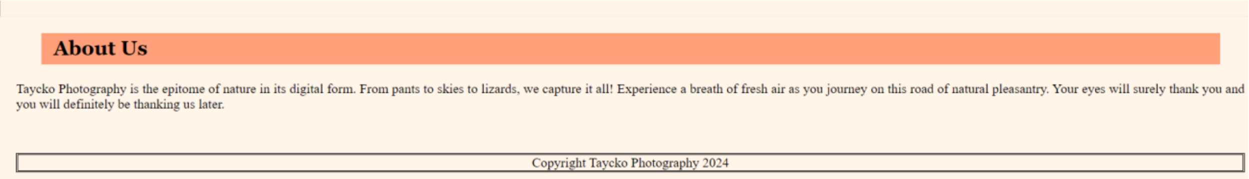


Want your photo on our site?

Sign up here!

Name Name
Email Email

Figure 3: Sign-up form made using CSS styling

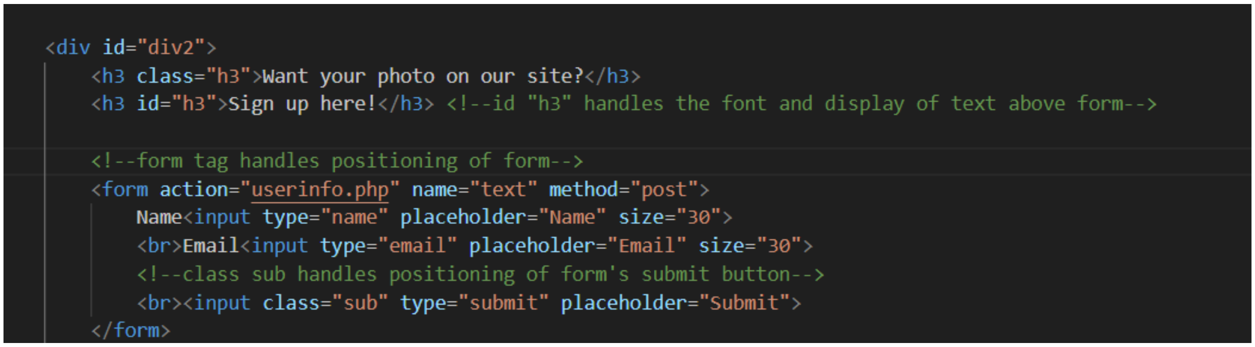


About Us

Taycko Photography is the epitome of nature in its digital form. From pants to skies to lizards, we capture it all! Experience a breath of fresh air as you journey on this road of natural pleasantry. Your eyes will surely thank you and you will definitely be thanking us later.

Copyright Taycko Photography 2024

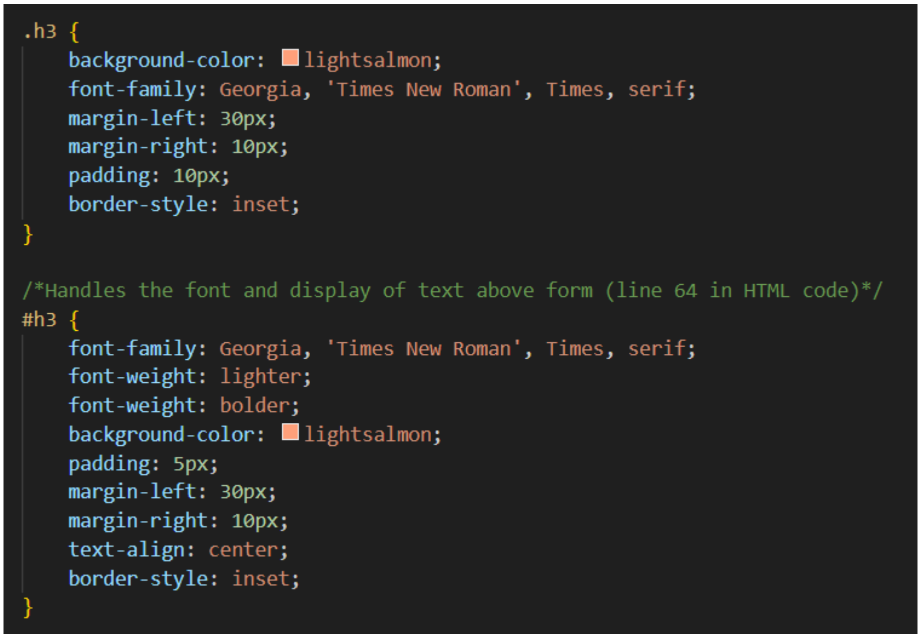
Figure 4: Company's About Us and copyright information



```html
<div id="div2">
    <h3 class="h3">Want your photo on our site?</h3>
    <h3 id="h3">Sign up here!</h3> <!--id "h3" handles the font and display of text above form-->

    <!--form tag handles positioning of form-->
    <form action="userinfo.php" name="text" method="post">
        Name<input type="name" placeholder="Name" size="30">
        <br>Email<input type="email" placeholder="Email" size="30">
        <!--class sub handles positioning of form's submit button-->
        <br><input class="sub" type="submit" placeholder="Submit">
    </form>
```

Figure 5: HTML code featuring the making of the sign-up form

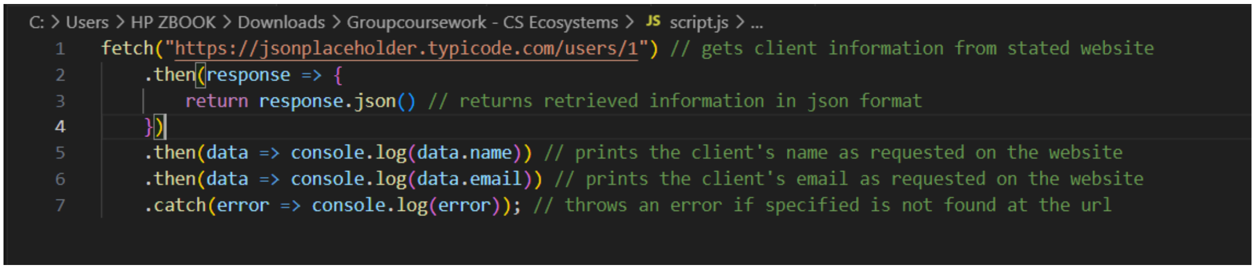


```css
.h3 {
    background-color: lightsalmon;
    font-family: Georgia, 'Times New Roman', Times, serif;
    margin-left: 30px;
    margin-right: 10px;
    padding: 10px;
    border-style: inset;
}

/*Handles the font and display of text above form (line 64 in HTML code)*/
#h3 {
    font-family: Georgia, 'Times New Roman', Times, serif;
    font-weight: lighter;
    font-weight: bolder;
    background-color: lightsalmon;
    padding: 5px;
    margin-left: 30px;
    margin-right: 10px;
    text-align: center;
    border-style: inset;
}
```

Figure 6: CSS code used in creation of the sign-up form

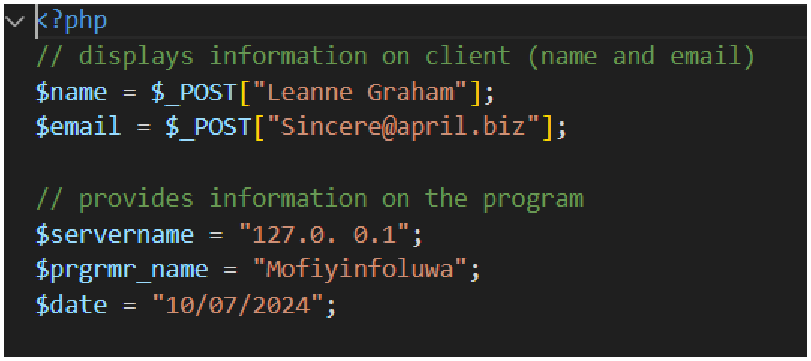
Moreover, a JavaScript file was implemented into the HTML code to collect the user’s sign-in information on the webpage. In particular, since the webpage demands the user’s name and email, only this information is displayed. Matter of fact, to visually see user information, we decided to add a PHP file that shows that the required data has been collected from the user.



C: > Users > HP ZBOOK > Downloads > Groupcoursework - CS Ecosystems > JS script.js > ...

```javascript
fetch("https://jsonplaceholder.typicode.com/users/1") // gets client information from stated website
    .then(response => {
        return response.json() // returns retrieved information in json format
    })
    .then(data => console.log(data.name)) // prints the client's name as requested on the website
    .then(data => console.log(data.email)) // prints the client's email as requested on the website
    .catch(error => console.log(error)); // throws an error if specified is not found at the url
```

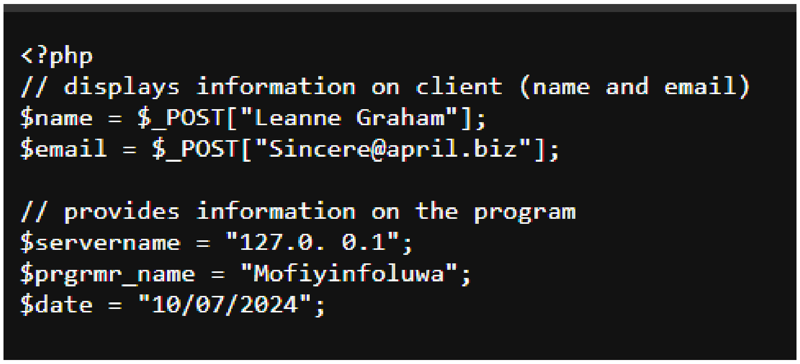
Figure 7: JavaScript Fetch API to collect user data



```php
<?php
// displays information on client (name and email)
$name = $_POST["Leanne Graham"];
$email = $_POST["Sincere@april.biz"];

// provides information on the program
$servername = "127.0. 0.1";
$prgrmr_name = "Mofiyinfoluwa";
$date = "10/07/2024";
```

Figure 8: PHP code to display user data



```
<?php
// displays information on client (name and email)
$name = $_POST["Leanne Graham"];
$email = $_POST["Sincere@april.biz"];

// provides information on the program
$servername = "127.0. 0.1";
$prgrmr_name = "Mofiyinfoluwa";
$date = "10/07/2024";
```

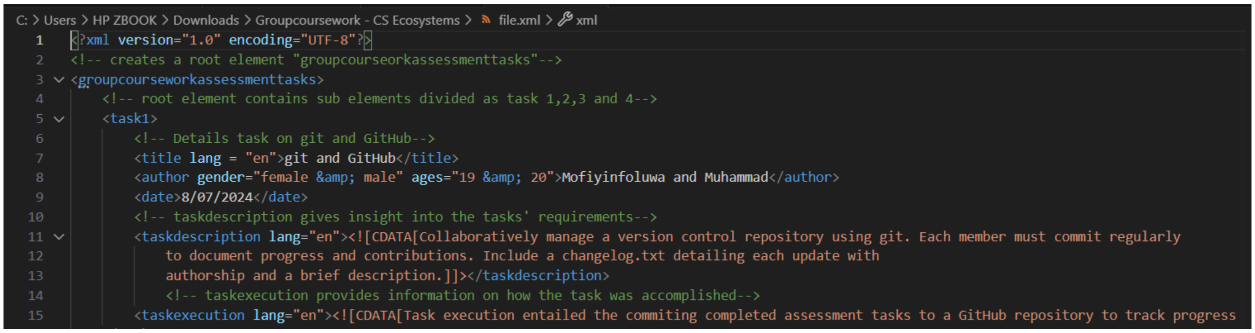
Figure 9: PHP-displayed user data

LaTeX

Our latex documentation was done focusing on structure and concise explanations of each task. After adding a document title, author names and date, the document was split into six categories including an Introduction, Git, HTML/CSS/JS, LaTeX, XML and a conclusion. Each category contains a short explanation of how each task was completed by both team members.

XML

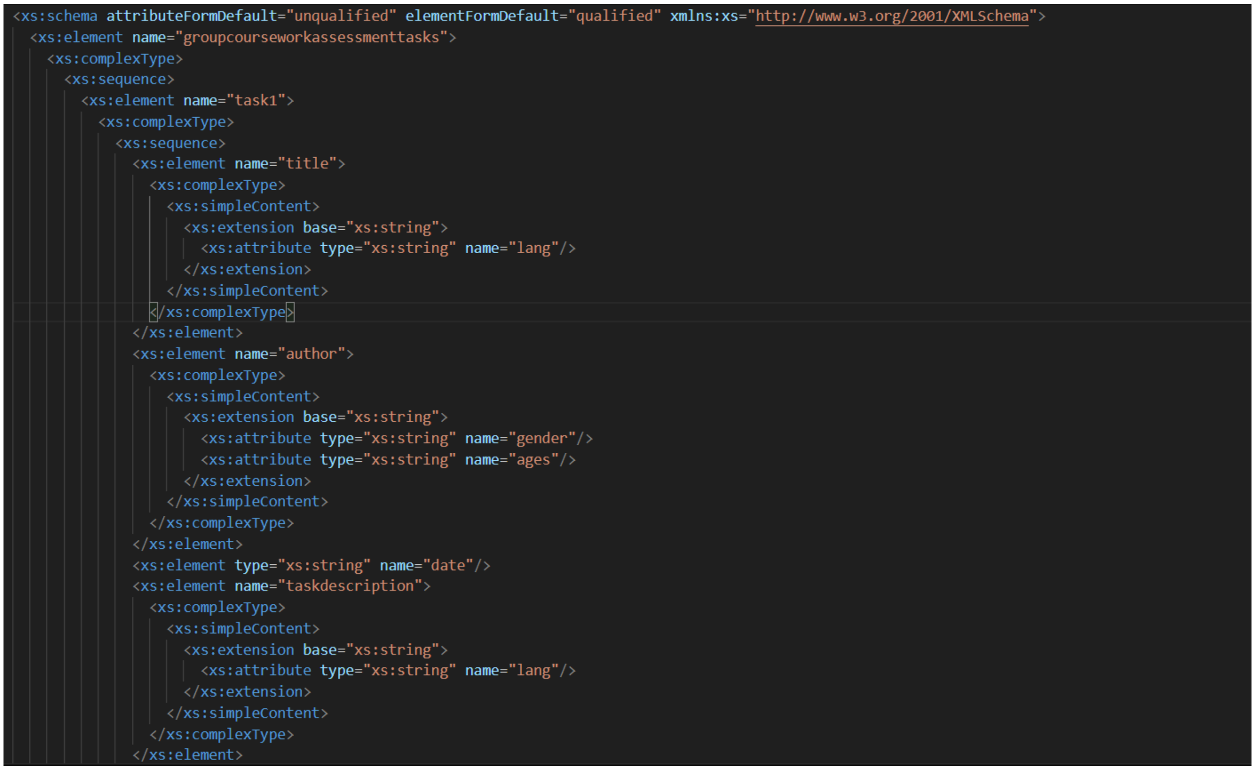
For XML, we decided to create a file that provides details on who did what throughout the assessment. Specifically, we created a root element “groupcourseworkassessmenttasks” with sub-elements task1, task2, task3 and task4. Within each of these subtasks, we provided information on the title of the task, the author, the date, the task description and the task execution. To provide more details about the assessment, attributes such lang, gender and age were used. In addition, to ensure that the XML code follows its set rules, an XSD schema and a validator were used to ensure that no errors ensued.



C: > Users > HP ZBOOK > Downloads > Groupcoursework - CS Ecosystems > file.xml > xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- creates a root element "groupcourseorkassessmenttasks"-->
<groupcourseworkassessmenttasks>
    <!-- root element contains sub elements divided as task 1,2,3 and 4-->
    <task1>
        <!-- Details task on git and GitHub-->
        <title lang = "en">git and GitHub</title>
        <author gender="female &amp; male" ages="19 &amp; 20">Mofiyinfoluwa and Muhammad</author>
        <date>8/07/2024</date>
        <!-- taskdescription gives insight into the tasks' requirements-->
        <taskdescription lang="en"><![CDATA[Collaboratively manage a version control repository using git. Each member must commit regularly
            to document progress and contributions. Include a changelog.txt detailing each update with
            authorship and a brief description.]]></taskdescription>
            <!-- taskexecution provides information on how the task was accomplished-->
        <taskexecution lang="en"><![CDATA[Task execution entailed the commiting completed assessment tasks to a GitHub repository to track progress
```

Figure 10: Example XML code featuring root element and its sub elements



```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="groupcourseworkassessmenttasks">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="task1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="lang"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="author">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="gender"/>
                      <xs:attribute type="xs:string" name="ages"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="date"/>
              <xs:element name="taskdescription">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="lang"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
```

Figure 11: XML XSD validation schema

Conclusion

In conclusion, this assessment brief required the implementation of a diverse range of Computer Science concepts. It helped us to build cooperation and organisational skills whilst also grounding our knowledge of the task's topics. Although considerably long, the tasks were moderately easy and exciting to execute. It is with utmost certainty that this coursework has helped us become more efficient programmers who have developed increased comprehension of different areas of Computer Science which we know is a necessity in the modern world.