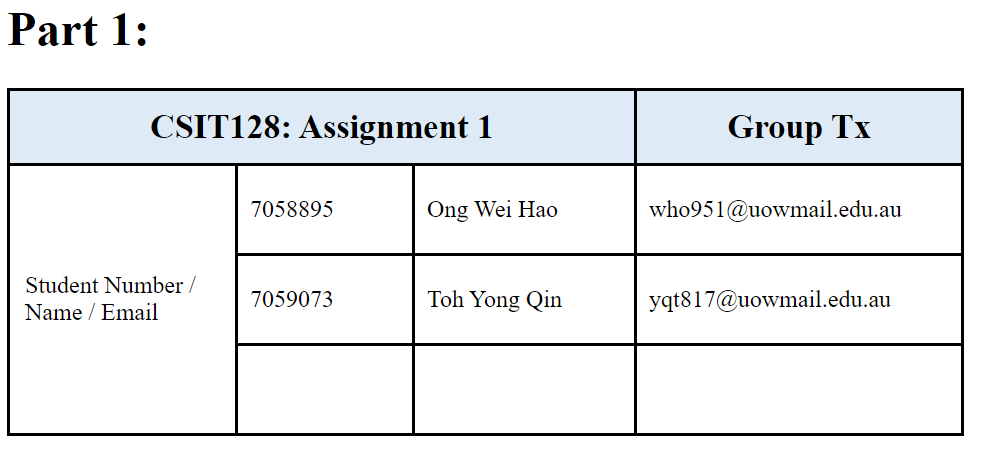
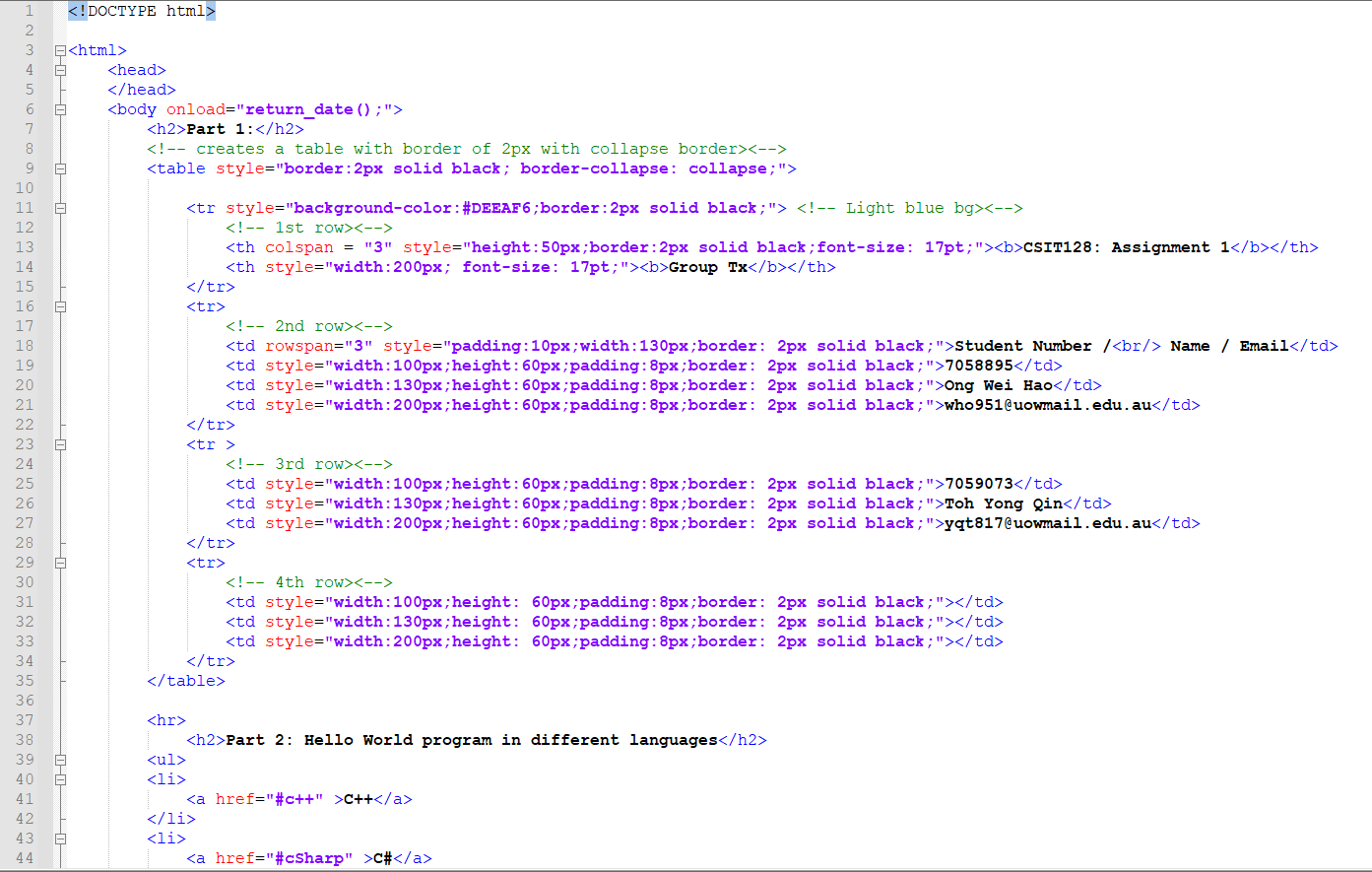
CSIT128: Assignment 1

Tutorial class and group: T214

|  |  |  |
| --- | --- | --- |
| Student Number: | 7058895 | 7059073 |
| Name: | Ong Wei Hao | Toh Yong Qin |
| Email: | who951@uowmail.edu.au | yqt817@uowmail.edu.au |

****

****

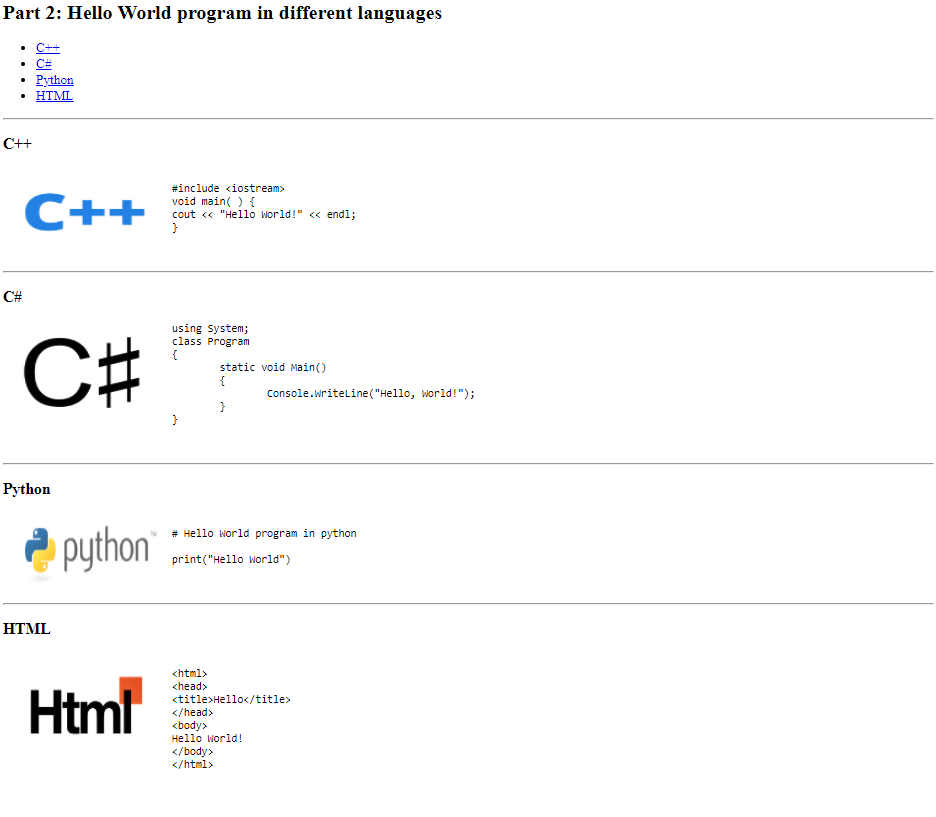
**Explanation:**

For <table> tag, on default, all cells of the table are separated meaning there is a gap between all cells. Setting border-collapse as collapse ensures no gaps in between each cell and also 2px solid black as the table border.

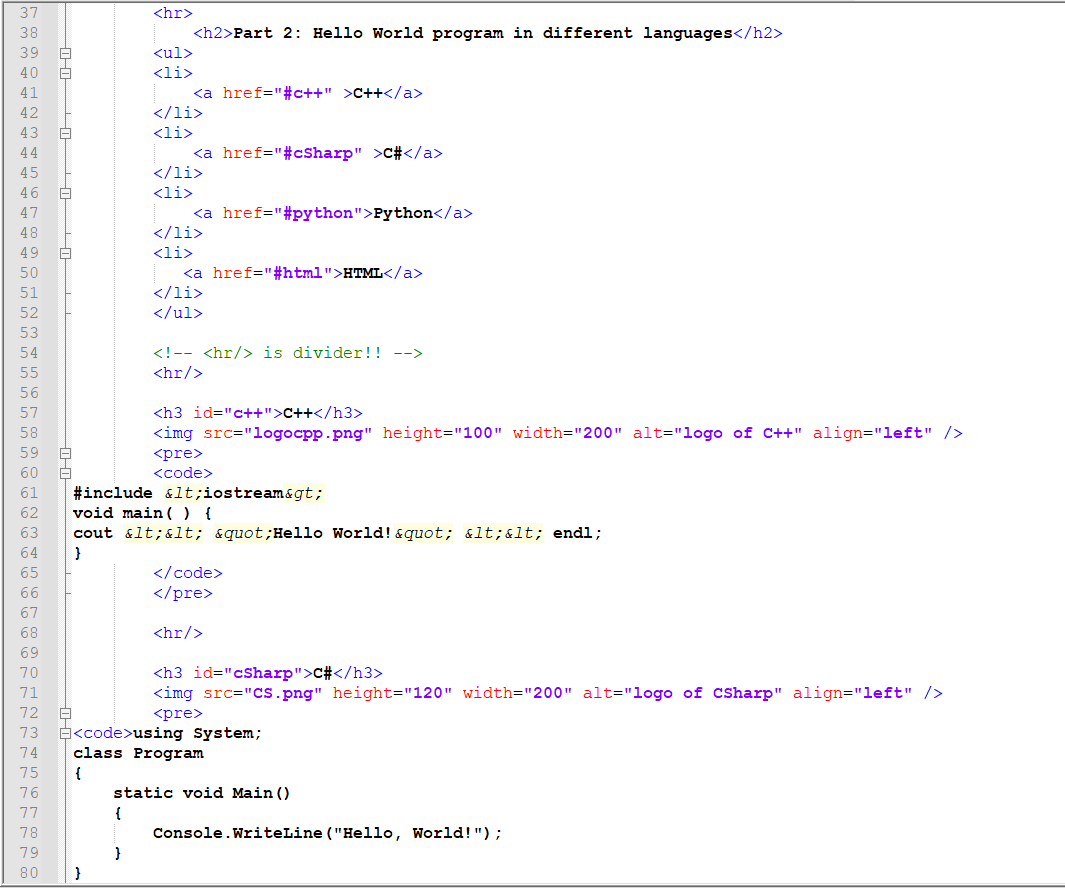
Moving on to the <tr> tag in the 1st row, our team has decided to set solid black 2px again individually for the cells because the previous one is solely for the table outline, whereas this only applies for the 1st row. Our team has also changed the background color to light blue for the first row similarly as what was depicted in the question. Next, for the first header with inner html "CSIT128:Assignment 1", as the cell covers 3 columns, setting colspan as 3 merges 3 columns into 1 column. <b> changes the wordings to bold and font-size to increase the font size, similarly for the 2nd header as well.

As for the 2nd row <td> tag, it consists of 3 rows. Setting rowspan as 3 combines 3 rows of the same column together. <br/> means the moving to the next line. Moving on to the last few cells, height and width are set as the same for the corresponding column to ensure that for cases where there is empty data in the cells, they will not be compressed down. Thus setting minimum width and height ensures consistency amongst the cells.

**Part 2**



**Code for part 2:**





**Explanation:**

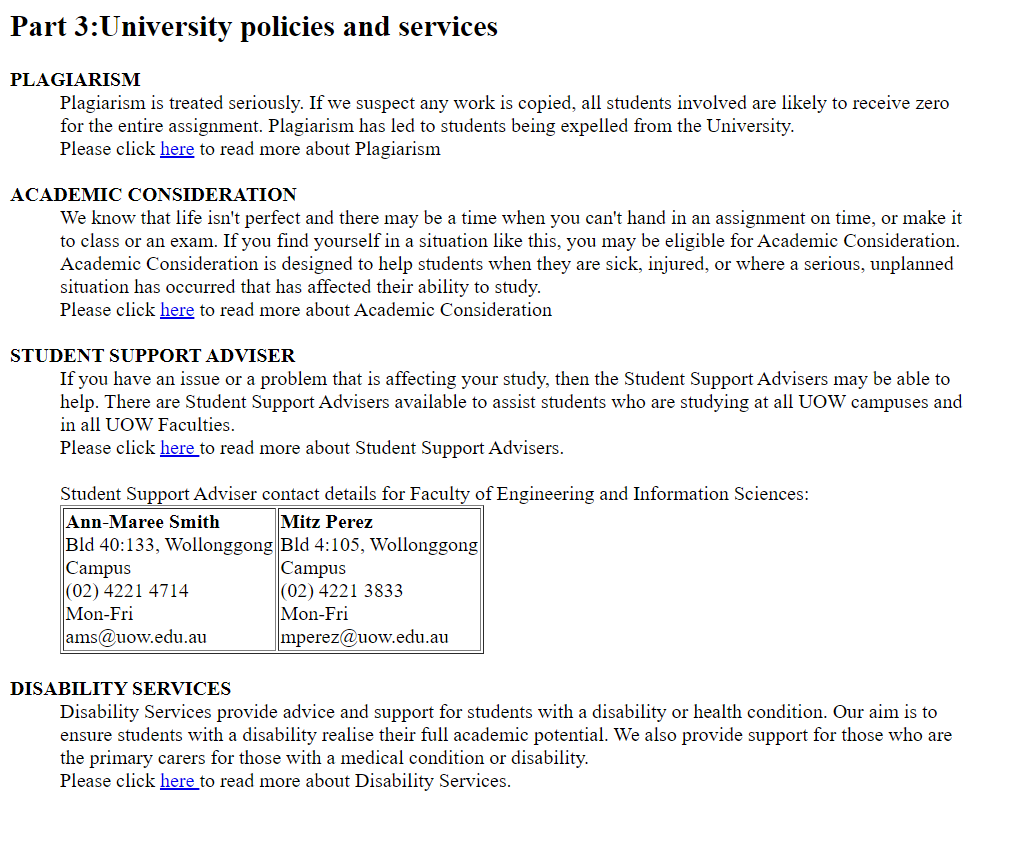
At the top part, there’s a heading which is done using the header tag of <h2>. Followed by bullet points, which is done by using the tags <ul> and <li>. The <a href> inside is the hyperlink. The href is set to the specific ids so when a user clicks on it, it will bring them to the specific part of the page that has that id.

</hr> is a divider which draws a horizontal line across the page. <img> tag is to display an image and the src is referenced to whatever the name of the image is. The alt inside displays whatever that is written if the image couldnt be found.

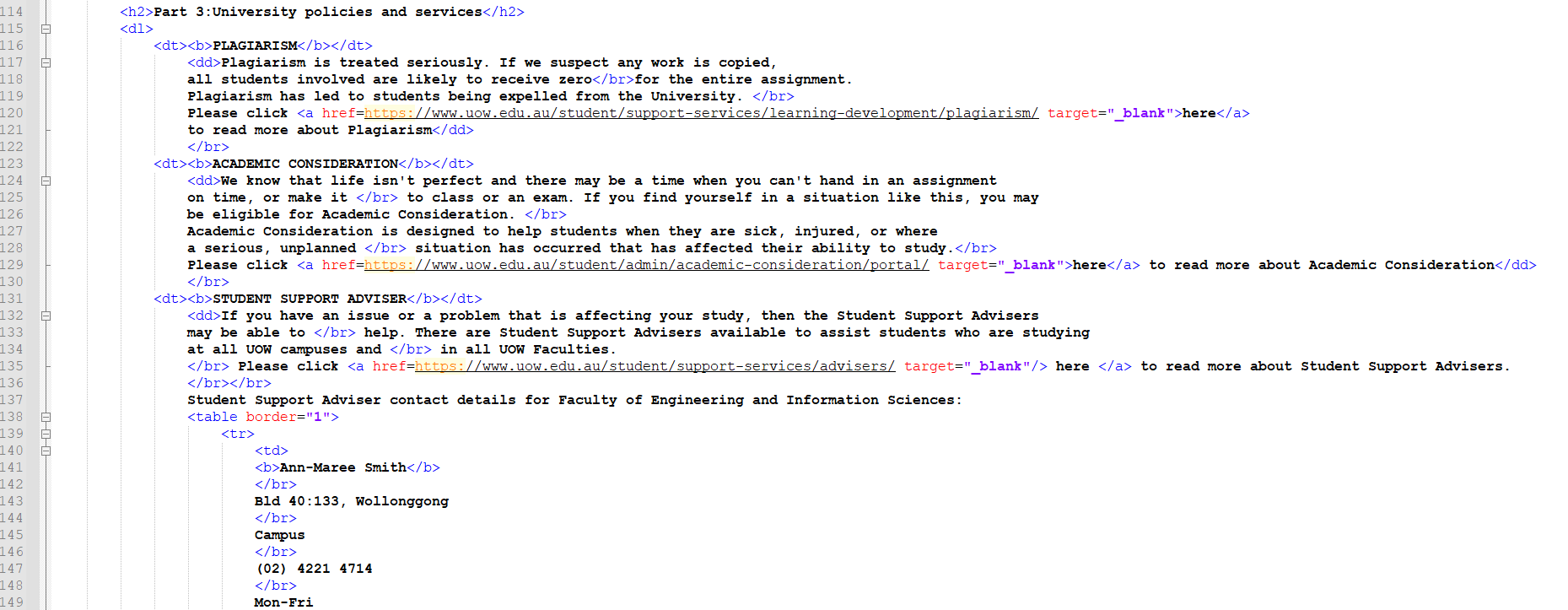
In this case, instead of using table format to display the image on the left and text on the right, I managed to figure out an alternative way to do it which is to align the image to the left by using align=”left”. By doing so, the image will then be on the left side and texts will be on the right.

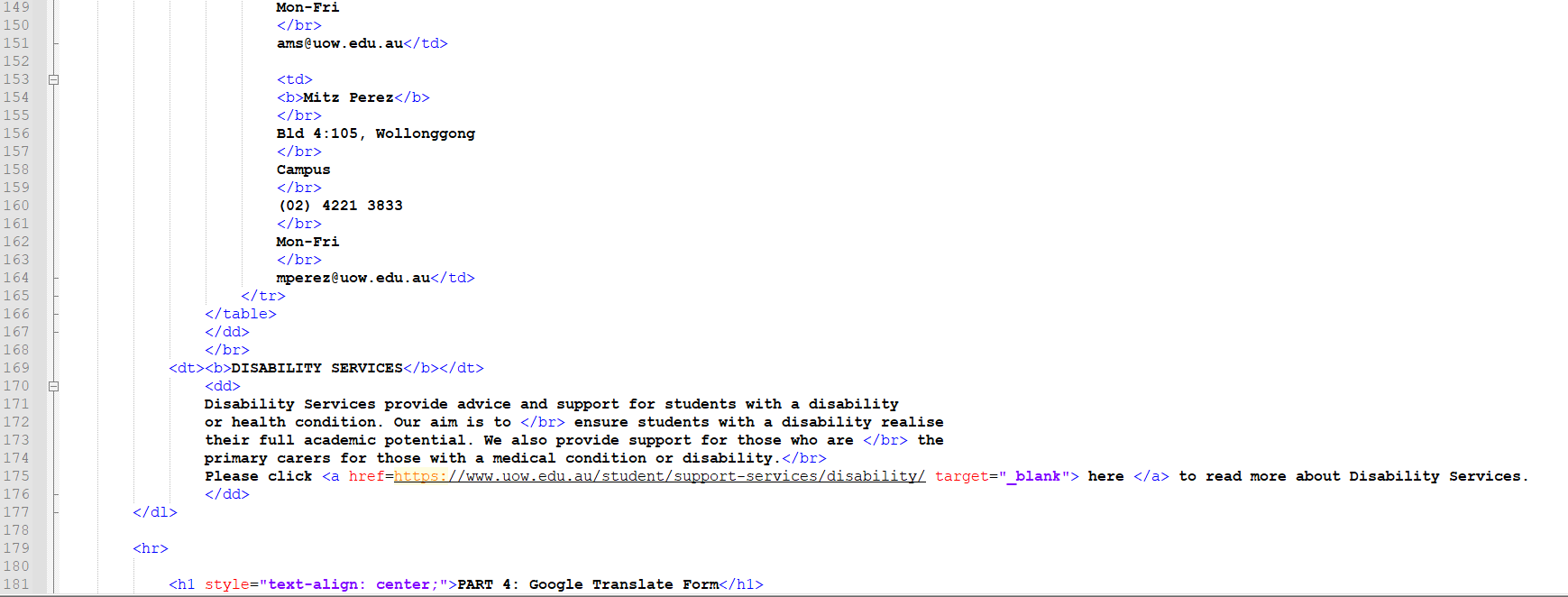
&lt;&gt;etc are all character entities and <pre> is to keep the code in the format and <code> is to let html know that we are typing codes.

**Part 3**



**Code for part 3:**





**Explanation:**

At the top part, there’s a heading which is done using the header tag of <h2>. <dd> and <dt> tag represents definition list.

First part is the part for plagiarism. <a> href is the hyperlink and it is set to UOW’s website and thus when a user clicks on it, it will bring them to the website via the link. Target is set to blank for the hyperlink so that when the user clicks on it, it will open a new tab after clicking on the link. Same goes for the academic part, student support advisor part and disability service. Table is constructed using <table> and inside there is <tr> which represents table data. </br> represents break.

**Part 4:**



**Code for name to replace text field of the form:**



**Explanation**:

Form id is set as "part\_4\_form", target='blank', action="https://translate.google.com" method="get" means that when i click translate button, a new tab opens and directs me to google translate website with all the names and values of the form inserted at the end of the url eg. https://translate.google.com/?name=value.

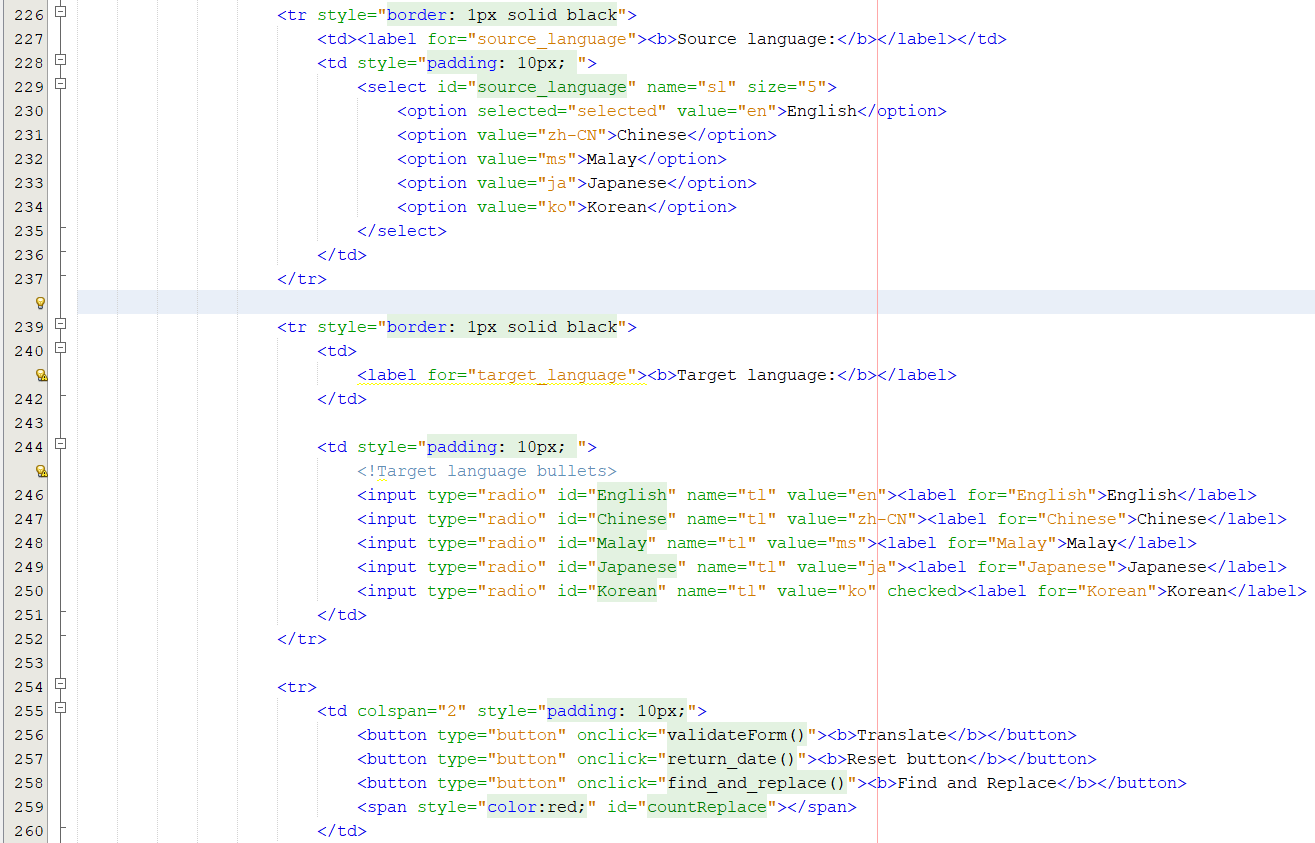
Next the <table> tag. Border-collapse as collapse makes cells compact with no gaps, margin auto left and right centers the table within the page. Next, the <label> tag is created with the words "name" as inner html. For the next column, a text field is created with value and name set as "name" to identify the text field. The textfield tag has an onchange listener event to ensure that the textfield is validated such that no special characters, numbers and blank fields are allowed. Next whenever the textfield is not properly filled, a message will prompt the user to warn them about their mistake through the span tag. This goes the same as the 2nd row of the table for Module Code.

As for the 3rd row of the table, current date, the text field associated with it is disabled by default so users will never be able to edit. The value in the text field will be auto generated by the system the moment the page is loaded up, through onload event listener in the <body> tag using the "return\_date()" function defined in <script>.

Moving on to the 4th row of the table, we have the message label and a textarea. The textarea consists of 3 rows and 20 columns. It has an onchange event listener using "validateMessage" where any empty field will prompt an error message at the <span> tag next to it.

As for the 5th row and 6th row, there are two labels which indicate find and replace, and two textfield for each label. The text field with id="replace" is coded such that on default is disabled and enabled the moment there is at least a single character in the input of the text field with id ="find". What makes this possible is the use of oninput event listeners using "enable\_disable\_replace()" function and will be explained down below.

**Code for source language to the end of the form:**



**Explanation:**

Next row, for the source language, we have the label and the selection list. The selection list contains english being the default, chinese, malay, japanese, korean language. The value of each language, and name are taken from the original google translate's html file. As for the target language, we have labels and bullet points instead. Each bullet point is bound with their respective labels. Also, the values are the same as the one we used for source language.

Next row, we have the buttons for translating, resetting and replacing. The types are all set as buttons which allows much more flexibility such that resetting and submitting of forms will be done in javascript.

**JAVASCRIPT:**

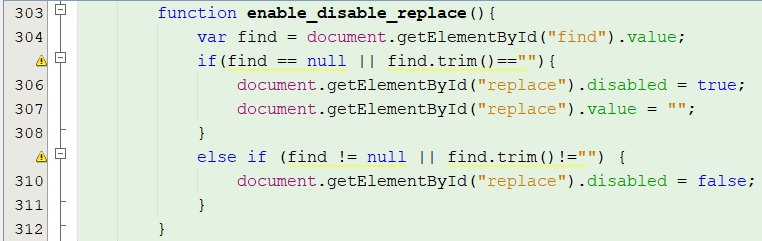
**function return\_date():**



**Explanation**:

First, we initialize an object from the Date class. Then, we obtain hours, minutes etc by passing through methods from the object and then concatenate them together to obtain the format that we wanted. Afterwards, we declare a form reset and then update the current date text field value. As we wanted the current date text field to be seen when the page loads up, we placed this method in the <body> tag as onload listener event. Also, i placed this method as onclick listener event for the reset button allows the form to reset before updating the current date text field, if not the current date text field gets erased every time the form resets.

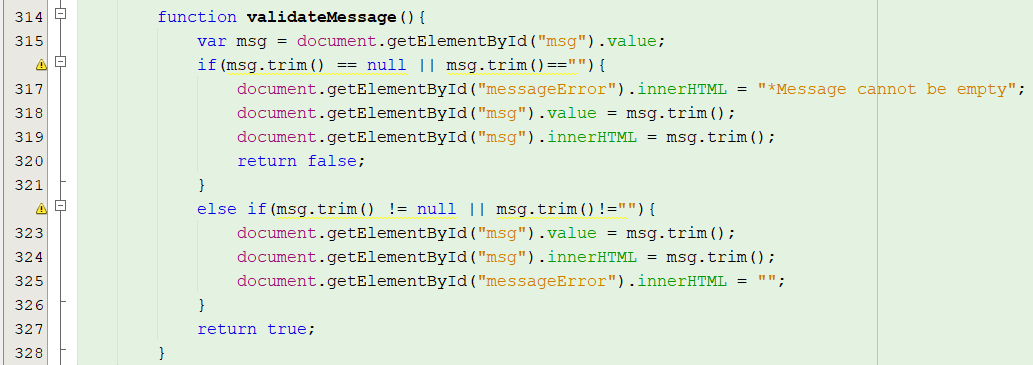
**Function for enabling or disabling the replace's text field:**



**Explanation**:

We obtain the value from find's text field first. If the find's text field is empty, replace's text field will be disabled and emptied, else replace's text field is enabled.

**Function for validating message:**



**Explanation:**

We obtain the value from message's text area. If the value is null or is a blank string after trimming, the span tag's inner html will change to "\*Message cannot be empty" and returns false, else the message will be trimmed and returned true.

**Junction for replacing message text area and calculate number of replaces:**



**Explanation:**

We obtain the value of id=" find", "replace", "msg". Run the split function to split up the string value of id="msg" by space, which turns into an array. Initialise a dummy counter variable as zero. If the value of id="find" is null or null after trimming, the inner html of <span> tag of id="countReplace" will become a blank string and returns false. So now, we iterate through the array by checking for similar values of the array to value of id="find", set the array's value as the replacing value and increments the counter variable if true. Next we join all the values of the array together while using space as a separator into a string variable. Also set the message's text area value as the newly joined string. If the counter is zero, we set the inner html of the span tag id="countReplace" as the counter itself, which will show just on the right of the "Find and replace" button.

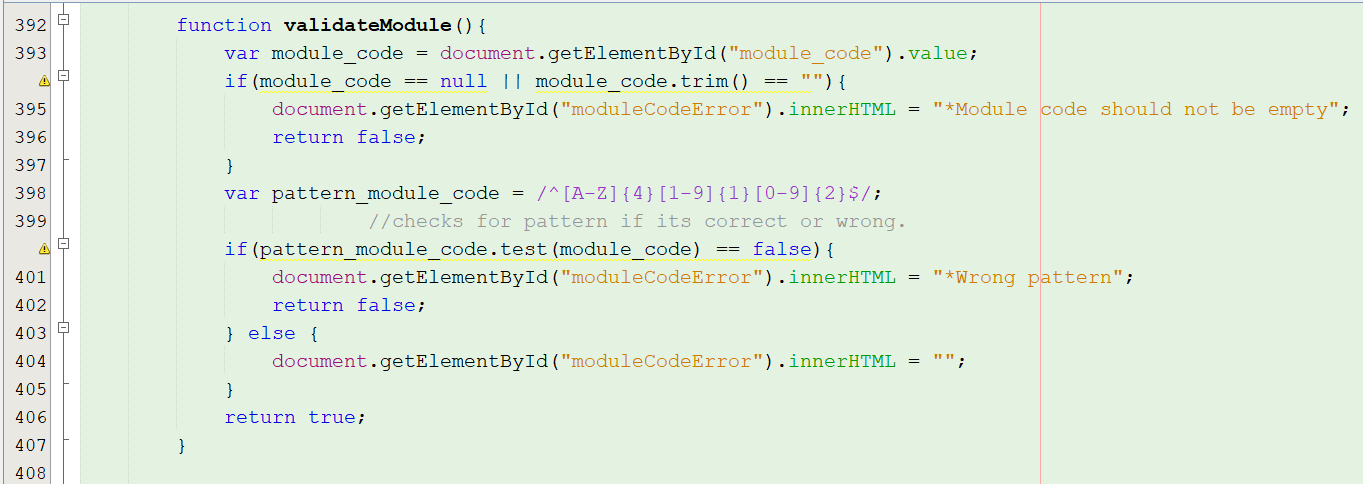
**Function for validating name:**



**Explanation**:

For validating names, we used regular expressions. If the name contains any special characters and digits and is empty, an error message will appear at the right of the text field.

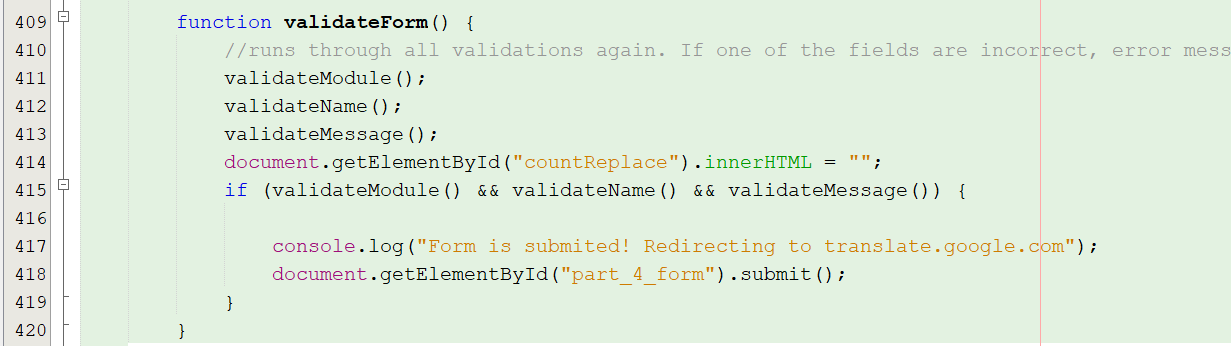
**Function for validating module code:**



**Explanation:**

For validating module code, we used regular expressions too. Error messages appear on the right side of the text field if they're incorrectly filled.

**Function for submitting form:**

****

**Explanation:**

For the last function, it runs through all validation functions again and submits the form if all validation functions return true, which then brings to the google translate website.