Assignment Instructions: Implementing HTML Putting Formatting Tags To Use

Objective

In this assignment, you are required to create a simple HTML document demonstrating the use of various formatting tags such as `<b>`, `<i>`, `<u>`, `<strong>`, and `<em>`. You will also be tested using a custom JavaScript test case to ensure that your HTML file meets the required structure and content.

Instructions

1. Implementing the HTML Putting Formatting Tags To Use

Implementing the HTML Putting Formatting Tags to Use

You will start with a blank `index.html` file. Follow the instructions below to implement the required HTML content:

1. HTML Structure: You need to create the basic structure of an HTML document.

2. Content: The document must include a main heading (`<h1>`), multiple paragraphs using formatting tags (`<b>`, `<i>`, `<u>`, `<strong>`, `<em>`), and a subheading (`<h2>`).

3. The HTML file should be structured as follows:

- The `<html>` tag must have a `lang='en'` attribute.

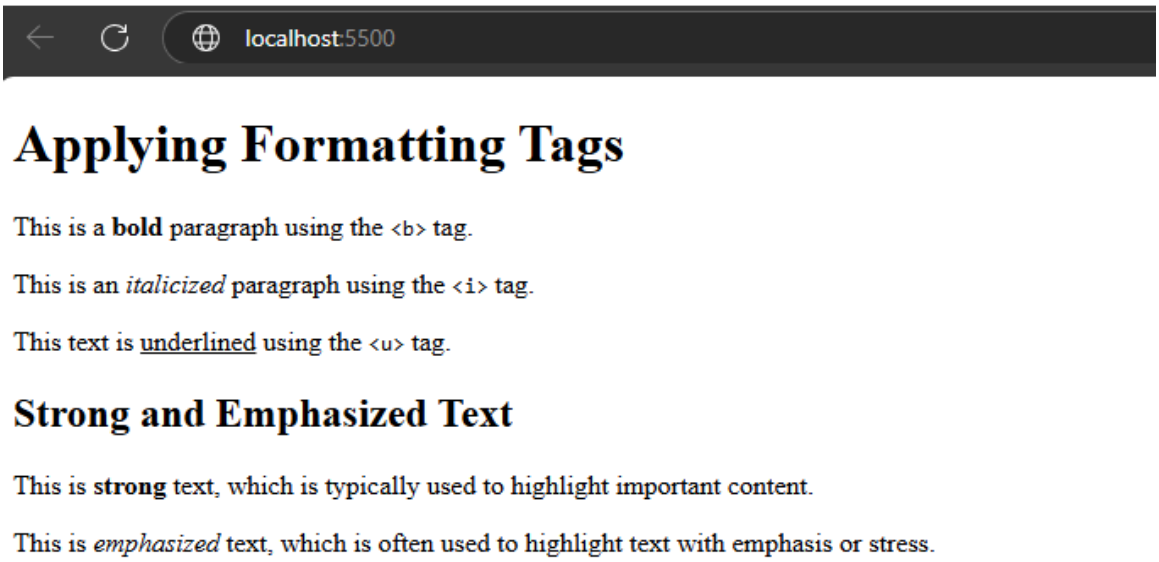
- The `<head>` section must contain a `<meta charset='UTF-8'>` tag and a `<meta name='viewport'>` tag for responsiveness.

- The `<title>` tag must be set to 'Putting Formatting Tags to Use'.

- The `<body>` section should contain one `<h1>` heading, multiple `<p>` paragraphs, and one `<h2>` subheading.

2. HTML Code: Putting Formatting Tags To Use

Here is the exact HTML code you need to implement inside your index.html file:



3. Explanation of the HTML Code

Here is a breakdown of the HTML code:

1. HTML Structure:

- Begin your HTML document with the `<!DOCTYPE html>` declaration, indicating that this is an HTML5 document.

- The `<html lang='en'>` tag wraps the entire content of your document, specifying that the language is English.

- Inside the `<head>` section, include the following:

  - A `<meta charset='UTF-8'>` tag, ensuring that the document uses UTF-8 character encoding.

  - A `<meta name='viewport' content='width=device-width, initial-scale=1.0'>` tag, making sure that the page is responsive and scales properly on different devices.

  - The `<title>` tag should be set to 'Putting Formatting Tags to Use', which will appear in the browser tab.

- Inside the `<body>` section, you need to add:

  - An `<h1>` tag should be present with text as 'Applying Formatting Tags'.

- Three `<p>` tags for paragraphs. Here's what each should contain:

    - The first `<p>` tag should demonstrate the use of the bold tag for bold text. The corresponding `<p>` tag should say: 'This is a **bold** paragraph using the <code><b></code> tag.'

    - The second `<p>` tag should demonstrate the use of the italic tag for italicized text. The corresponding `<p>` tag should say: 'This is an *italicized* paragraph using the <code><i></code> tag.'

    - The third `<p>` tag should demonstrate the use of the underline tag for underlined text. The corresponding `<p>` tag should say: 'This text is <u>underlined</u> using the <code><u></code> tag.'

- An `<h2>` subheading should be present with text as 'Strong and Emphasized Text'. Inside this section, add:

    - A `<p>` tag demonstrating the use of the strong tag for strong emphasis. The corresponding `<p>` tag should say: 'This is **strong** text, which is typically used to highlight important content.'

    - A `<p>` tag demonstrating the use of the em tag for emphasized text. The corresponding `<p>` tag should say: 'This is *emphasized* text, which is often used to highlight text with emphasis or stress.'

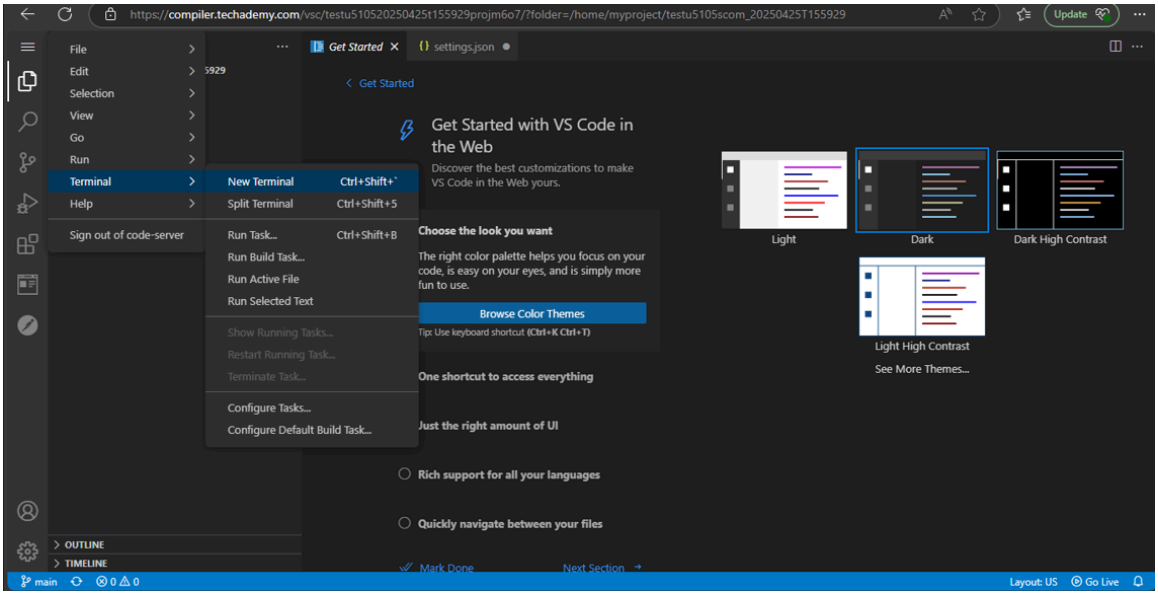
Assessment Guidelines

Step 1:

- Once the VS Code interface loads in the browser, wait until you see the workspace and left sidebar.
- To open the command terminal the test takers, need to go to

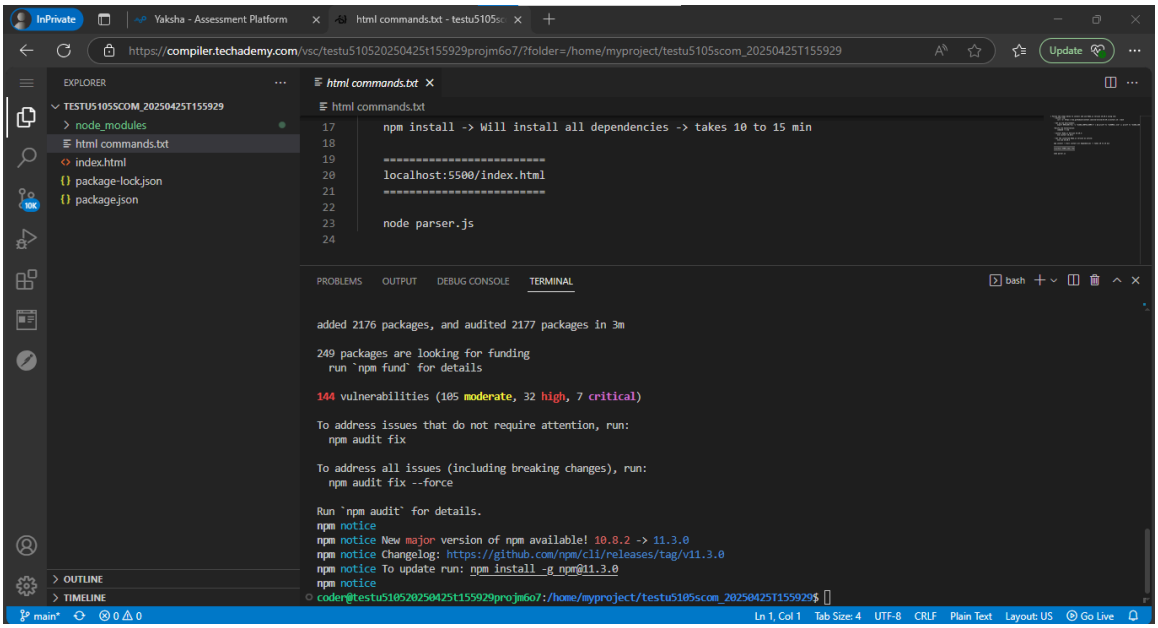
  Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.

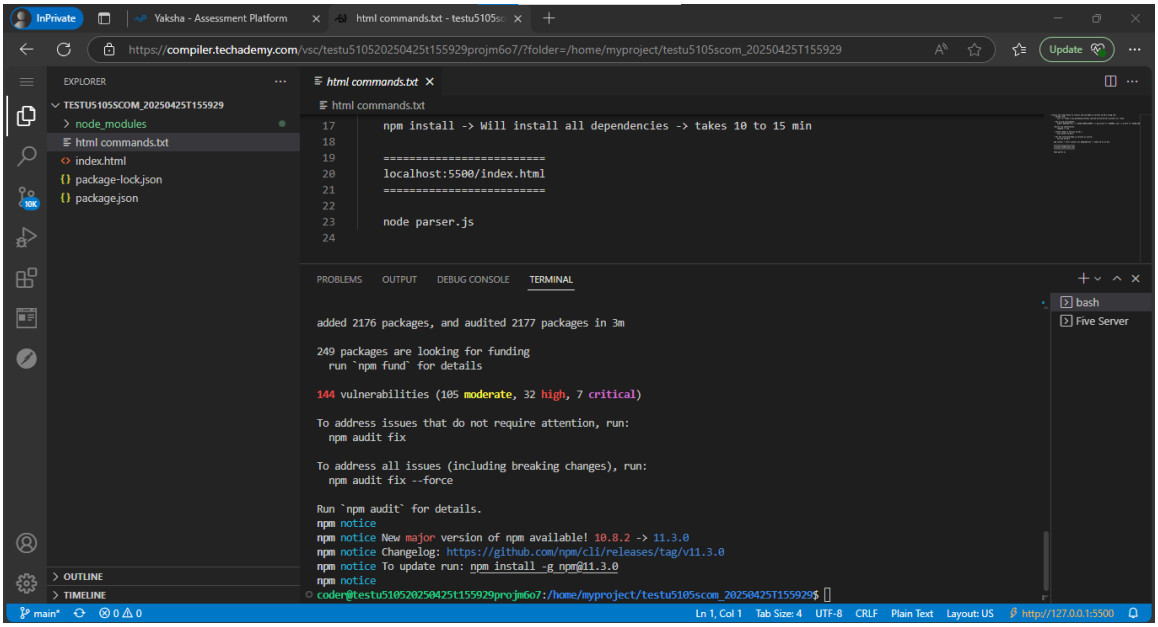
Now in the terminal you need to install all dependencies using the “npm install --no-bin-links --unsafe-perm” command.



Step 2:

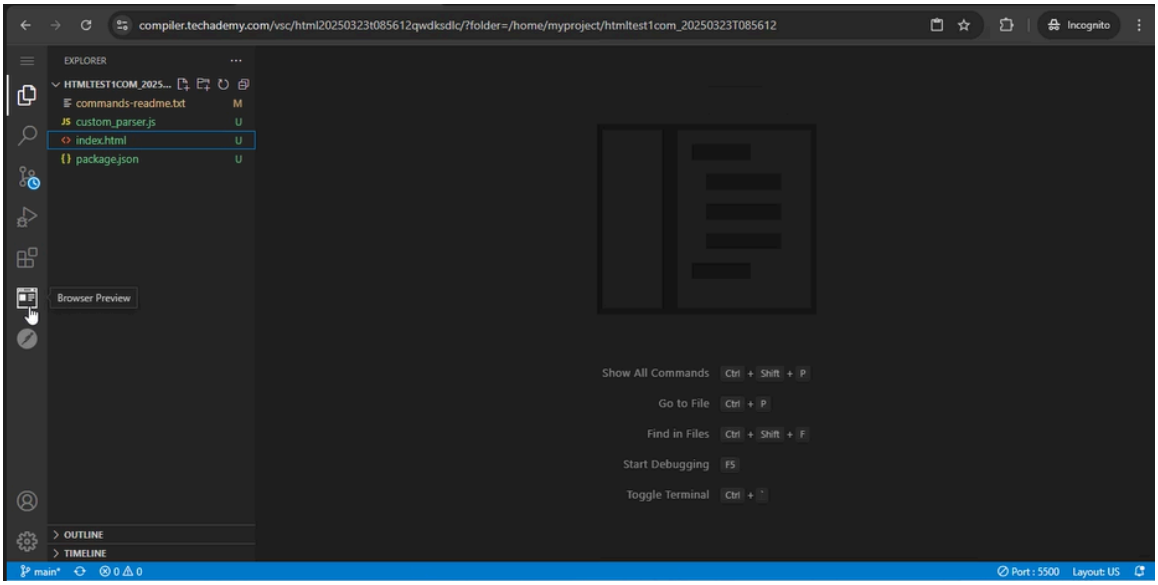
- Once installation completes, go to the **bottom right corner** of the VS Code screen.
- Click the **"Go Live"** button – This will start a **live server**, The server will run at port `5500` (e.g., `http://localhost:5500/`)





Step 3: Preview Output in Browser

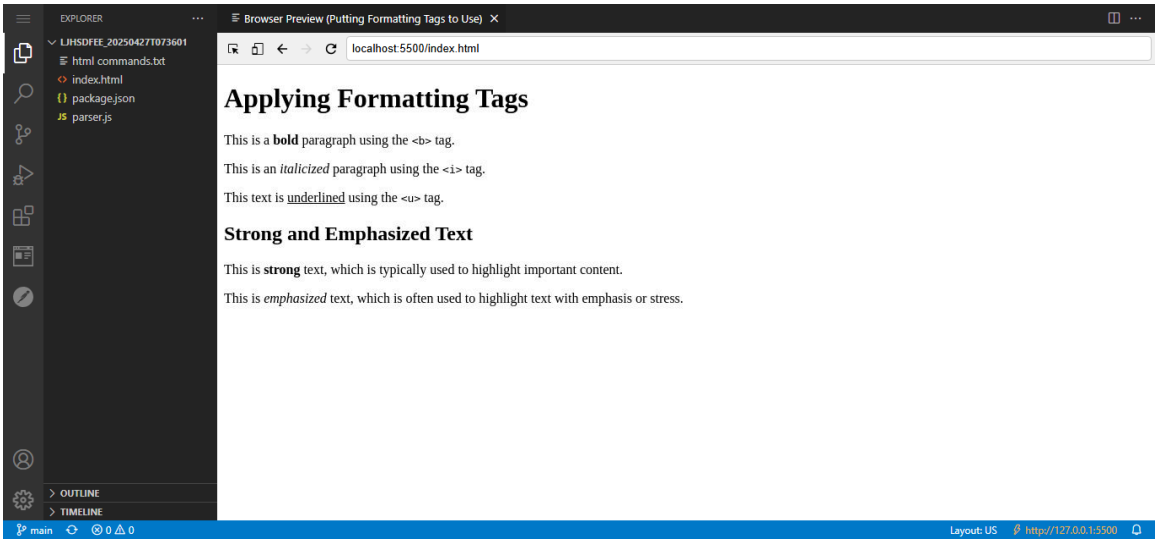
- This is a **web-based application**, so to view it in a browser, use the **internal browser inside the workspace**.
- Click on the **second last icon on the left panel** (the one labeled **"Browser Preview"**). This will open a tab within VS Code where you can **launch and view your application**.
- Note: The application **will not open in your system's local browser** — it must be viewed using the internal browser.



In the **Browser Preview tab**, type the following URL in the address bar and press **Enter**:

Your file is being served on: `localhost:5500/index.html`

This will load your HTML file and display the output of your web page **inside the internal browser**.



Step 4:

- Go back to the **terminal** and type the following command, then press **Enter**:

  node parser.js

- This command will **execute the validation script** and display the test results for your HTML file in the terminal.

Mandatory Assessment Guidelines:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save**

the updated contents in the internal git/repository. Else the code will not be available in the next login.

5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

   Note: The application will not run in the local browser
7. You can follow series of command to setup HTML environment once you are in your project-name folder:
   a. npm install --no-bin-links --unsafe-perm -> Will install all dependencies -> takes 10 to 15 min.
   b. localhost:5500/index.html -> This will load your HTML file and display the output of your web page inside the internal browser.
   c. node parser.js -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.
8. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
9. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.
10. If `Ctrl + Shift + B` doesn't work, then manually run the following commands one by one in the terminal:
    ➔ git add .
    ➔ git commit -m "Final commit"
    ➔ git push