Web Systems and Technologies (EIE4432)
Lab 1: HTML and CSS Basics

Expected Outcomes

- Create a functional and visually appealing user interface for a drink ordering system
- Showcase featured drinks and provide navigation option on homepage
- Allow users to select a drink, specify the size of the drink using HTML form
- Accurately represent the designed user interface and ensure an intuitive and enjoyable experience for users

Submissions

- Please refer to the teaching schedule for the deadline
- This is an individual work
- Students should host their work on a designated hosting service and provide a public domain for viewing the website live
- Students need to submit a report with screen captures showing they have achieved the outcomes stated above
- Every screen capture should be full screen and show the **Student ID, System Date & Time**, otherwise **1 grade would be deducted** from the laboratory
- Late submission would deduct 20-50% of the mark of this laboratory

- Please declare the use of GenAI tools and how they have been used in the appendix page similar to the following:

    - I declare that Generative AI tools have been used to prepare the submitted work. The Generative AI tools used and the manner in which they were used are as follows:
      _____.

    - You must reference them in according with the accepted academic conventions (e.g. IEEE or APA styles).

Assessment Criteria

The report will be assessed according to

- The clarity of presentation,
- Technical correctness, and
- Whether the screen captures can demonstrate that the outcomes have been achieved.

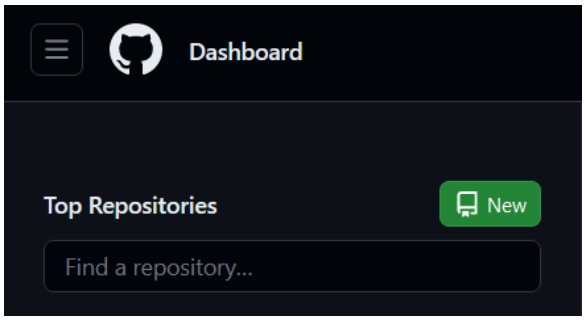
Lab Activities

- Development Environment & Project Setup (10 mins)
- Task 1: Homepage of Drink Ordering System (100 mins)
- Task 2: Drink Ordering Form (60 mins)
- Publish your Website (10 mins)

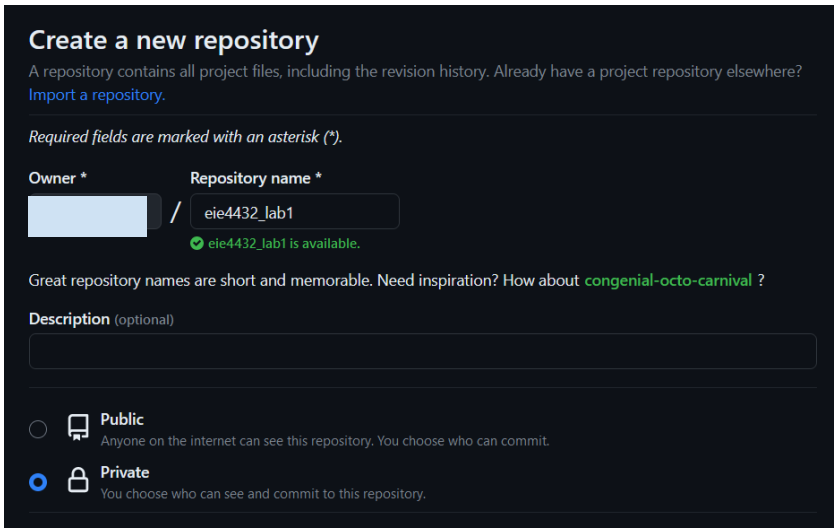
Development Environment & Project Setup (10 mins)

1. Go to GitHub (https://github.com/) and create an account by clicking “Sign up”
    - You may use your own account if you have already own one

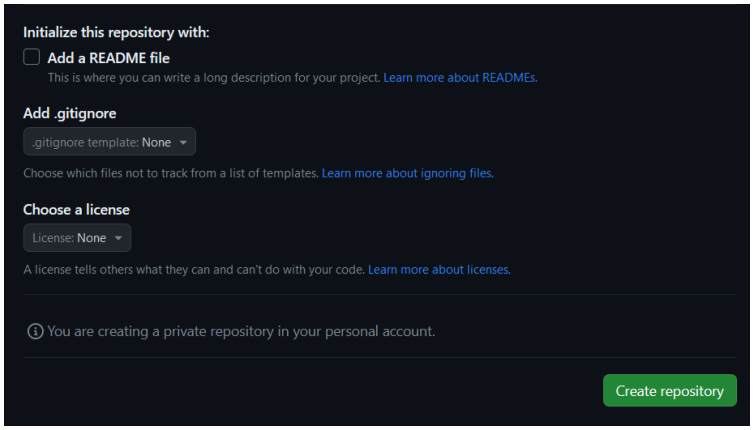
2. Enter your email, password & username, then click “Continue”
    - Your password should be at least 15 characters OR 8 characters including number and a lowercase letter
    - Then follow the steps to create an account on GitHub
3. Upon account registration, you will be redirected to dashboard of your account. Click “New” on left sidebar to create a new repository



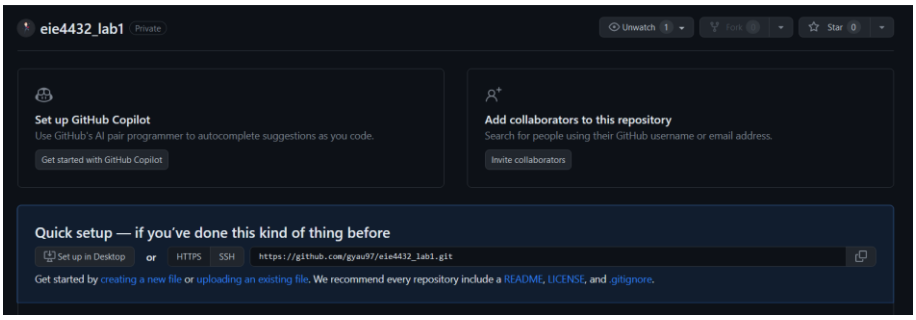
4. Enter “Repository Name” as “eie4432_lab1”, select “**Private**” for repository type
    - Repository: contains all your project's files and each file's revision history
    - **Public** Repository: accessible to everyone on the internet
    - **Private** Repository: only accessible to you, people you explicitly share access with, and, for organization repositories, certain organization members
    - Owner can always update repository visibility after creation



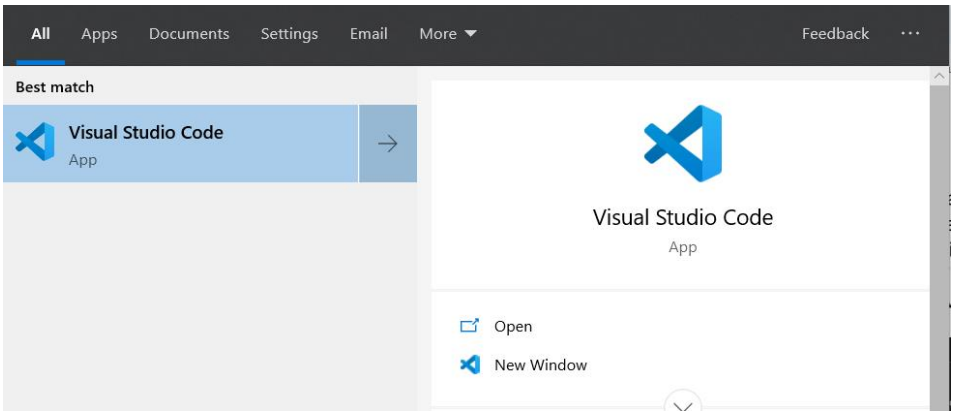
5. Keep original settings, click “Create Repository”
    - README file: provide an overview of the project and guide, such as installation procedures, introduction, usage, configuration, contribution list, license & credits
    - .gitignore: a text file used in Git repositories to specify files and directories that should be ignored and not tracked by Git



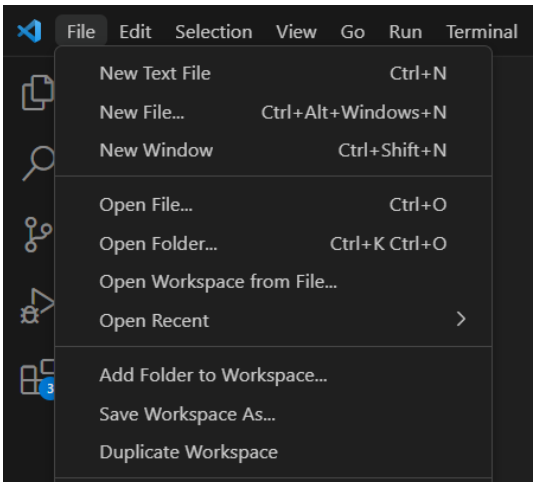
6. Repository is created successfully if following screen is shown



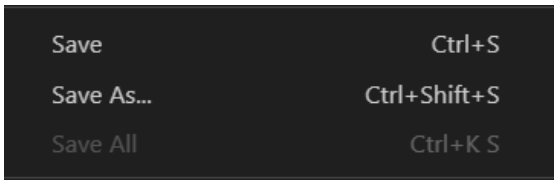
7. Start Visual Studio Code (VS Code) or any integrated development editor (IDE)
    - Press windows key, search “vs code” on search bar
    - Click “Visual Studio Code” to start the application



8. Click “File” on top left corner, choose “New Text File…”
    - Shortcut Key: (Windows) Ctrl + N (Mac) Cmd + N



9. A new tab of “Untitled-1” should pop up, click “File” on top-left corner again, choose “Save as…”
    - Shortcut Key: (Windows) Ctrl + S (Mac) Cmd + S



10. Create a folder on designated location, such as desktop, name it as “<Student ID>_EIE4432_LAB1”
11. Rename file as “index.html”, save it in the folder created

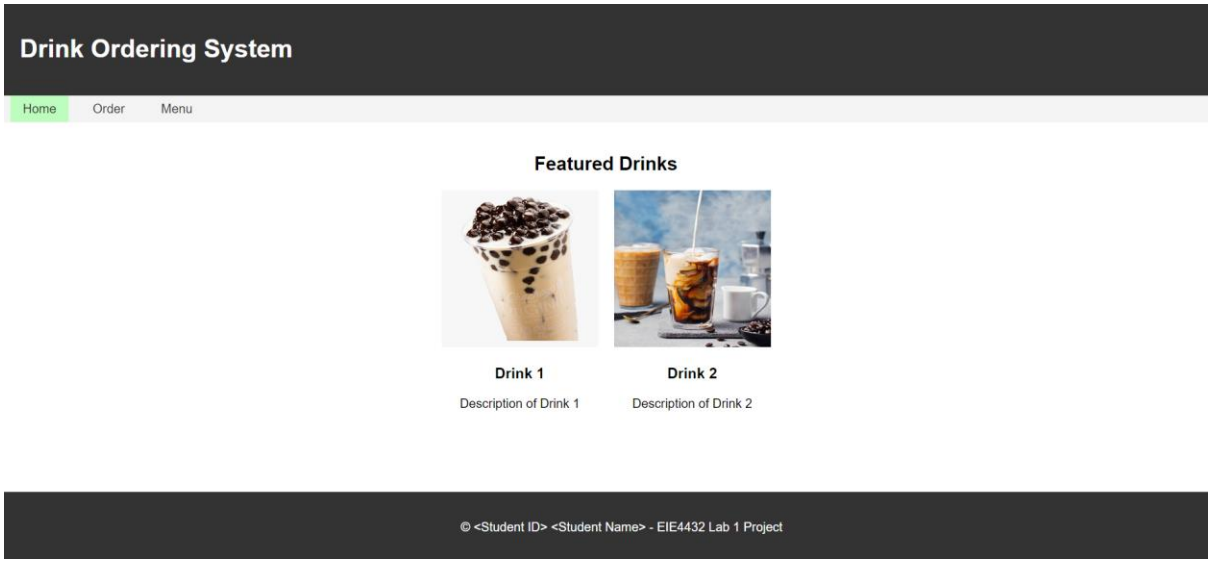
Suggested File Structure

```
└── <Student ID>_EIE4432_LAB1 (Project Root Folder)
    ├── styles
    │   ├── main.css
    │   └── order.css
    ├── assets
    │   ├── img1.jpg
    │   ├── img2.jpg
    │   └── etc...
    ├── index.html
    └── order.html
```

- styles folder: for organizing **CSS** files
- assets folder: for organizing all **multimedia** files, such as photos, videos, documents, etc…

Task 1: Homepage of Drink Ordering System (100 mins)

Expected Outcome



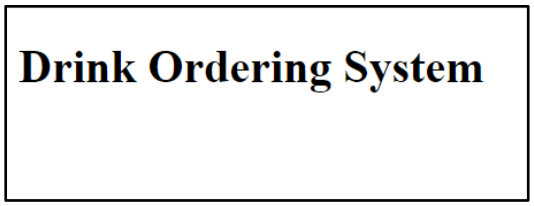
1. The webpage should have a title and utilize a CSS file for styling
2. The header should display the name of the drink ordering system
3. The navigation bar should provide links to the Home, Order, and Menu pages
4. The main section should showcase featured drinks
5. Each featured drink should have an image, a name, and a description
6. The footer should display the copyright information for the project

Instructions

1. Go to “index.html” and setup the HTML structure
2. Start with the <!DOCTYPE html> declaration to specify the HTML version
3. Create the <html> element as the root element
4. Inside the <html> element, add the <head> and <body> elements.
5. Go to <head> element, add <title> element, use “<Student ID>_Drink Ordering System” as page title
6. Save the file
   - Shortcut key: (Windows) Ctrl + S (Mac) Cmd + S
7. Find the location of the saved HTML file, double click to open it with web browser

Lab #1.1
Provide a screenshot of your code & the browser screen with your **Student ID and System Date & Time**, which should be showing the text of <title> element.
Your code should be indented correctly (each nested tag should be indented exactly once inside of its parent tag).

8. Go to <body> element, create the <header> element to encapsulate the header content
    - <header> is a HTML5 element represents a container for introductory content or a set of navigational links
9. Use the <h1> element to display the title of the drink ordering system
    - Refresh the page, a page title is shown



10. Below <header> element, use <nav> element to create navigation links
    - <nav> element acts as a container for the navigation content
11. Add <ul> element in <nav> element, and 3 list items
    - Usually, a set of navigation links is a list structure to hold the navigation items, and use CSS to stylize for its navigational appearance
12. Inside each <li> element, there is an <a> element representing a navigation link
    - 1st link: Home
    - 2nd link: Order
    - 3rd link: Menu
    - For each <a> element, it should have the "href" attribute to hold an anchor or hyperlink. At this moment, there is no link yet, place "#" as a placeholder
    - Save & refresh the browser page, the following is shown

Lab #1.2
Provide a screenshot of your code & browser screen with your **Student ID and System Date & Time**, which should be showing the page title and an unordered list of navigation links.

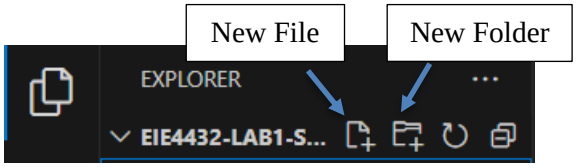
13. Below <nav> element, use <main> element to encapsulate the header content, create a <section> element inside
    - In <main> element, there are few more semantic elements for website structure, these semantic tags are helpful and clearly describes its meaning to both the browser and the developer
    - <section>: defines a section in a document
    - Reference: https://www.w3schools.com/html/html5_semantic_elements.asp
14. In <section> element, display a sub-heading for "Featured Drinks" using <h2> element

15. Create a element to represent a drink container, which would contain an image, name of the drink, and its pricing
    - There are 2 drinks for featured drinks
    - We always use element to represent a container for content, by adding class name(s) could make it more clear on what is containing in the
        - e.g.
16. Save “bubble-milktea.png” and “iced-latte.jpg” into “assets” folder, you can find it from the **eie4432-lab1.zip**
17. In element, use <img> element to display a drink image
    - Reference: https://www.w3schools.com/tags/tag_img.asp / Refer to Lecture 1
    - 1st drink: “bubble-milktea.png”
    - 2nd drink: “iced-latte.jpg”
    - alt & title attributes are required
        - 1st drink: Bubble Milktea
        - 2nd drink: Iced Latte
18. Following the image, display the name of the drink using <h3>
    - 1st drink: Bubble Milktea
    - 2nd drink: Iced Latte
19. Lastly, there is a <p> element containing the drink's price information, each price should use 1 line to display
    - 1st drink: $30 (M) $35 (L)
    - 2nd drink: $30 (M) $35 (L)
    - Save & refresh the browser page and see the output on browser

Correct Display	Incorrect Display
Bubble Milktea $30 (M) $35 (L)	**Bubble Milktea** $30 (M) $35 (L)

Lab #1.3
Provide a few of the screenshots of browser screen with your **Student ID and System Date & Time**, which should be showing the “Featured Drinks” sub-heading and 2 drink images with their information.

20. According to the suggested file structure, in the root project folder (<Student ID>_EIE4432_LAB1), create a new folder “styles”, create a new file and rename it as “main.css”
    - Either create it in ordinary way in File Explorer
    - Or create it in VS Code, which is like how you create “index.html” file

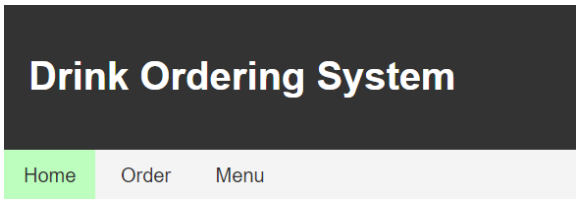


21. In <head> element, add the following code to include the "main.css" file
    - <link rel="stylesheet" href="styles/main.css">
    - Ensure that the "styles" folder is present in the same directory as the "index.html" file
22. Open “main.css” file from “styles” folder in VS Code
23. Add the following code for styling on <body> element
    - body { }
    - In the curly brackets, press an Enter and start giving CSS properties for styling
24. Specifies the font family to be used for the text within the body to Arial, also add a fallback **sans-serif** font to ensure legibility
    - font-family: Arial, sans-serif;
    - Always end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available
    - font-family: https://www.w3schools.com/cssref/pr_font_font-family.php
25. Remove margins and padding from the body, which helps to removing any default spacing around the content
    - margin: https://www.w3schools.com/cssref/pr_margin.php
    - padding: https://www.w3schools.com/cssref/pr_padding.php
    - Margin & Padding control spacing around content
26. Apply flex container properties to the body using display properties and set to flex
    - display: https://www.w3schools.com/cssref/pr_class_display.php
    - Turns the body element into a flex container, allowing its children to be flex items
    - This enables flexible layout options
27. Add the following code to specify that the flex items should be arranged vertically, creating a column layout
    - flex-direction: column;
    - flex-direction: https://www.w3schools.com/cssref/css3_pr_flex-direction.php
28. To ensure the body occupies at least the full height of the viewport, sets the minimum height of the body element to 100% of the viewport height (vh)
    - min-height: https://www.w3schools.com/cssref/pr_dim_min-height.php
29. Save the changes made to the "main.css" file & refresh the “index.html” on browser

30. Stylize the <header> element with following requirements:
    - Create a new selector of CSS rules for <header> element
    - Set the background color of the header to a specific color using hex code
    - Set the text color within the header to white
    - Apply padding to the header for 20 pixels on all sides
    - Save the changes made to the "main.css" file & refresh the “index.html” on browser

> **Lab #1.4**
> Provide screenshots of your CSS code & browser screen with your **Student ID and System Date & Time**, which should be showing the color update on <header> part.

31. Stylize the navigation bar as following expected outcome



32. Add a new selector for the unordered list in <nav> element with following code:
    - nav ul {}
33. Remove the default list style from the navigation bar using list-style-type property
    - No bullet points should be seen
    - list-style-type: https://www.w3schools.com/cssref/pr_list-style-type.php
34. Remove margins and set padding for the navigation bar
    - For padding, set 12 pixels for top & bottom sides, also remove padding for left & right sides
35. Set a background color for the navigation bar
36. Add a new selector for items inside the unordered list, which responsible for nav items styling
    - nav ul li {}
37. Display the navigation items in a horizontal line by setting display property to “inline”
38. Apply padding to the navigation items
    - 12 pixels of padding on the top and bottom
    - 16 pixels of padding on the left and right
39. Add a new selector for navigation links
    - nav ul li a {}
40. Remove underlines from the navigation links by setting text-decoration property to “none”
41. Set the text color for the navigation links

42. Add a new selector for adjusting style when mouse is put on any navigation items (hover effect)
    - nav ul li:hover {}
    - Set the background color of the navigation item to a specific color represented by the hex code
43. Go to “index.html”, add a class attribute “currentPage” to “Home” navigation item
44. Back to “main.css”, add a new selector for the current page in the navigation
    - li.currentPage {}
    - In CSS, “#” is used to represent an id, “.” is used to represent a class
    - So li.currentPage represents a <li> item with class currentPage
45. Set a different background color for current page by hex code
46. Save the changes made to the "main.css" file
47. Add a new selector for main section
    - main {}
48. Set the main section to flex and adjust its size using flex property and set to 1
    - It allows the main section to expand and fill the available space within its parent container
    - Control flex-grow, flex-shrink, flex-basis properties at the same time
    - flex: 1 is used to distribute the available space equally among flex items within a flex container
    - flex: 0 is used to prevent flex items from growing or shrinking
    - flex: https://www.w3schools.com/cssref/css3_pr_flex.php
49. Apply padding to the main section for 20 pixels on all sides
50. Add a class attribute to the <h2> element (Featured Drinks) in “index.html”
    - class: section-header
51. Back to “main.css”, add a new selector for section header
    - .section-header {}
52. Center align the text “Featured Drinks”
53. Add a class attribute to the element under <h2> in “index.html”
    - class: featured-drinks
54. Back to “main.css”, add a new selector for section header
    - .featured-drinks {}
55. Display the featured drinks in a flex container and center align them
    - Use display property and set it to “flex”. It allows items inside this to be displayed in a row
    - Also, use “justify-content” property and set it to “center”. It centers the featured drinks within the flex container
56. Add a class attribute to the elements nested inside “featured-drink” class in “index.html”
    - class: drink-container
57. Back to “main.css”, add a new selector for section header
    - .drink-container {}
58. Display the drink containers in a flex layout using display property, allowing them to be aligned horizontally
59. Add a class attribute to the element nested inside “drink-container” class in “index.html”

- class: drink

60. Back to “main.css”, add a new selector for section header
    - .drink {}
61. Apply margin and center align the drink elements
    - 0 pixels of margin on the top and bottom
    - 10 pixels of margin on the left and right
    - Use text-align property to make text centered inside the
62. As images are inside element with “drink” class, no need to specify class for images
63. Add a new selector for section header
    - .drink img {}
64. Set a specific width and height for the drink images using width & height property
    - 200 pixels for both width and height
65. Save the changes made to the "main.css" file & refresh the “index.html” on browser

Lab #1.5
Provide screenshots of your CSS code & browser screen with your **Student ID and System Date & Time**, which should be showing all <main> elements. Every element should be centered, 2 drinks information are horizontally aligned.

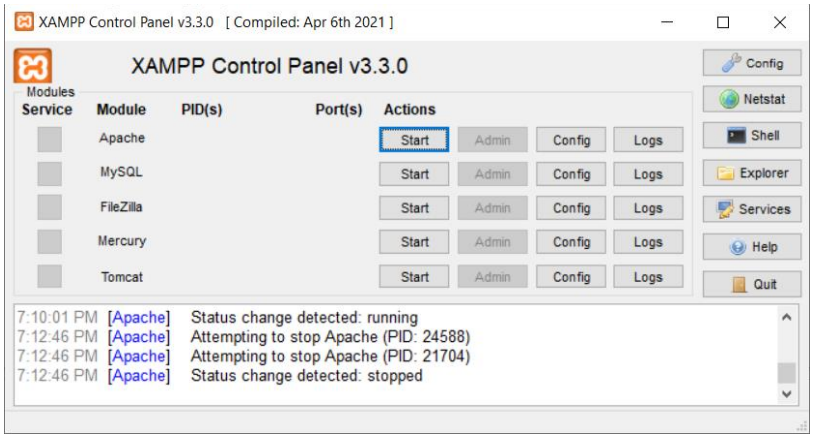
66. Back to “main.css”, add a new selector for footer
    - footer {}
67. Set the background color & text color of the footer, it should be the same as header, to keep the UI consistency
68. Apply padding to the footer
    - 20 pixels for all sides
69. Center align the text within the footer
70. Position the footer at the bottom of the page using position property and set it to “fixed”
    - It can position the footer relative to the viewport, ensuring it remains fixed even when scrolling
    - position: https://www.w3schools.com/cssref/pr_class_position.php
71. Set it stick to bottom and left sides of the browser by using “bottom” & “left” properties, and set to 0, so it can be aligned to bottom & left of viewport
72. Set the width of the footer to 100% of the viewport, ensuring it spans the entire width of the page
73. Save the changes made to the "main.css" file & refresh the “index.html” on browser

Lab #1.6
Provide screenshots of your CSS code & browser screen with your **Student ID and System Date & Time**, which should be showing all <footer> elements, with your student ID.

Before task 2, you are advised to run your code on localhost as testing as well.

Method 1: XAMPP

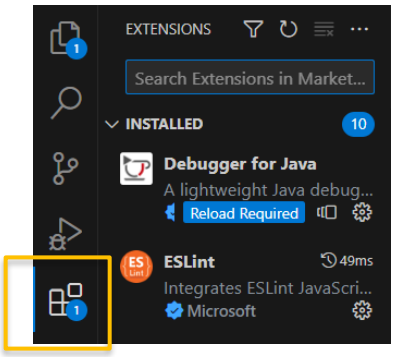
1. Open “xampp”, you can search in the window menu
    - Download link: https://www.apachefriends.org/
2. A control panel will be opened, click “Start” for “Apache”



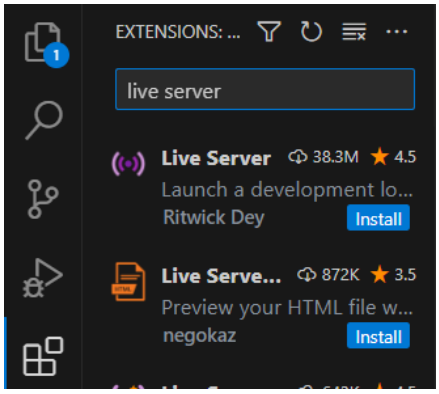
3. Click “Explorer” on the right, it will locate you the location of “xampp” folder on your computer
4. Open “htdocs” folder, create a new folder “eie4432_lab1”, and put your project files into the folder
5. Open browser, type “localhost/eie4432_lab1” on address bar
    - Format: localhost/<folder name>
    - It is automatically directed to “index.html”, if you do not include it in the folder, type your own file name, eg. “localhost/eie4432_lab1/home.html”

Method 2: Live Server Extension in VS Code

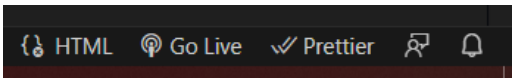
1. Click the icon at bottom of VS Code left side-bar



2. Search for “Live Server” by Ritwick Dey, and click “Install”



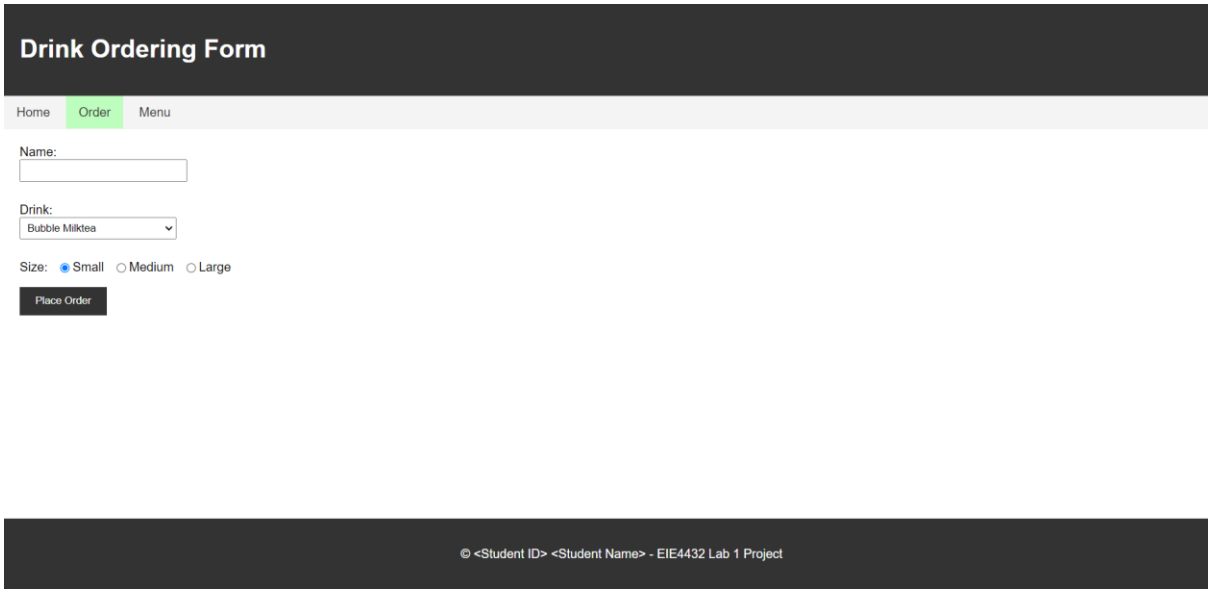
3. After installation, go back to the file editor (top icon of left side bar), click “Go Live” at bottom-right corner



4. The browser will open “localhost:5500” (or any port available on your computer)
5. You can see the changes directly upon file save

Task 2: Drink Ordering Form (60 mins)

Expected Outcome



1. The webpage should have a title and utilize a CSS file for styling
2. The header should display the name of the drink ordering form
3. The navigation bar should provide links to the Home, Order, and Menu pages
4. The main section should show the drink ordering form with appropriate input elements
5. A button for submitting the ordering form
6. The footer should display the copyright information for the project

Instructions

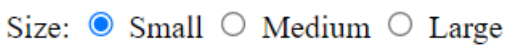
1. Go to “order.html” and setup the HTML structure as “index.html”
2. Go to <head> element, add <title> element, use “<Student ID>_Drink Ordering Form” as page title
3. Save the file
4. Find the location of the saved HTML file, double click to open it with web browser

> **Lab #2.1**
> Provide a screenshot of your code & the browser screen with your **Student ID and System Date & Time**, which should be showing the text of <title> element.
> Your code should be indented correctly (each nested tag should be indented exactly once inside of its parent tag).

5. Go to <body> element, create the <header>, <nav> and <footer> elements as “index.html”, you should keep the style is consistent as homepage
6. Create a <main> element below <nav> element
7. Add <form> element inside <main>, add “#” as placeholder for “action” attribute, choose “POST” as request method of this form

8. Within the form, create a element with the class "name-input"
9. Create a label and text input field in the class “name-input” element
    - Label: “Name:”
    - Type of input field: text
    - Make sure the input field is a required field
    - Set the id & name attributes as “name”

10. Create another element with the class "drink-select" for adding the Drink selection dropdown
11. Insert a <label> element with the "for" attribute set as "drink" and the text "Drink:"
12. Include a <select> element with the id attribute set as "drink" and the name attribute set as "drink"
13. Within the <select> element, add <option> elements for each drink option
    - Set the value attribute for each <option> element to correspond to the drink selection
    - Provide the drink name as the text content for each <option> element



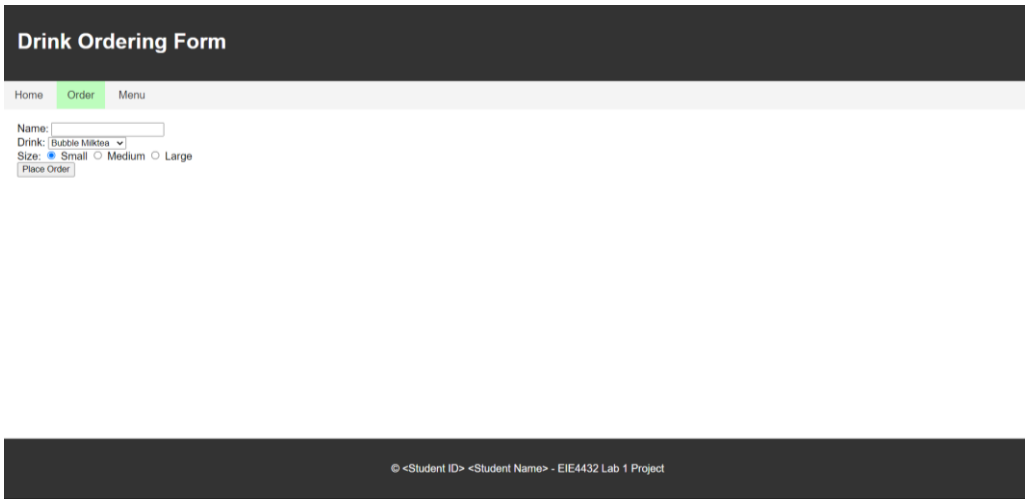
14. Within the form, create another element with the class "size-select"
15. Insert a <label> element with the "for" attribute set as "size" and the text "Size:"
16. Include 3 <input> elements with the type set as "radio"
17. Assign each <input> element a unique id attribute (e.g., "small", "medium", "large") and the name attribute set as "size"
18. Set the value attribute for each radio button to represent the corresponding size option
19. Use <label> elements with the "for" attribute matching the respective input's id to label each radio button

Place Order

20. Inside the form, create another element with the class "submission"
21. Include an <input> element with the type set as "submit" and the value set as "Place Order"

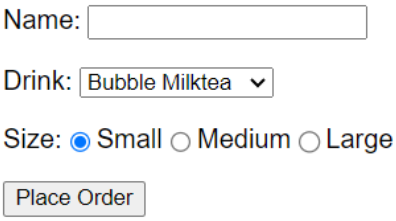
> **Lab #2.2**
> Provide a screenshot of your code & the browser screen with your **Student ID and System Date & Time**, which should be showing the whole web form of drink ordering.

22. Create "order.css" file in “styles” folder as creating “main.css” file (Refer to Task 1 Step 20)
23. In <head> element, add the following code to include the "order.css" file
    - <link rel="stylesheet" href="styles/order.css">
    - Ensure that the "styles" folder is present in the same directory as the "index.html" file
24. Open “order.css” file from “styles” folder in VS Code
25. Stylize the <body>, <header> and <nav> element as “index.html” in “order.css”
    - You should be able to reuse the CSS code from “main.css”, as the UI design between pages should be consistent
26. Go to “order.html”, add class attribute to “Order” navigation item, set it as “currentPage”
    - Make sure there is only 1 navigation item is associated with “currentPage” class

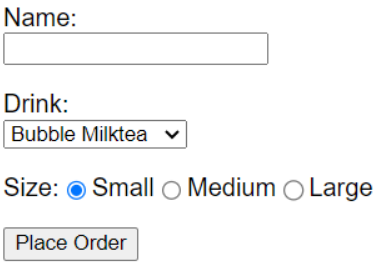


27. Stylize the <main> & <footer> element as “index.html” in “order.css”

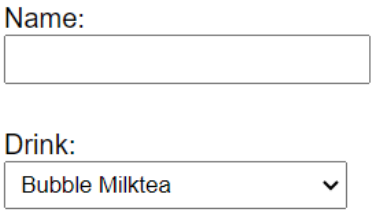
28. Add a new selector for <form> element
    - form {}
29. Apply 20 pixels of margin to the bottom of form to create spacing
30. Create a flexible layout for the form using “display” property and set it to “flex”
31. Stack the form elements vertically by setting “flex-direction” property to “column”
    - flex-direction: https://www.w3schools.com/cssref/css3_pr_flex-direction.php
32. Add a gap of 16 pixels between form elements to create spacing
    - gap: https://www.w3schools.com/cssref/css3_pr_gap.php



33. Add a new selector for the element with the “size-select” class
    - .size-select {}
34. Make the element having a flexible layout, so elements can be aligned horizontally and more packed

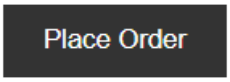


35. Add a new selector for all <label> elements
    - label {}
36. Set the “display” property to “block” to ensure each label is displayed on its own line



37. Add a new selector to target text input and select elements
    - input[type="text"], select {}
38. Set the width of the input and select elements to 200 pixels
39. Add 5 pixels of padding to the input and select elements
40. Add 10 – 20 pixels of margin at the bottom of the input and select elements to create spacing

41. Add a new selector to target radio button elements
    - input[type="radio"] {}
42. Add a left margin of 16 pixels to create spacing between each option



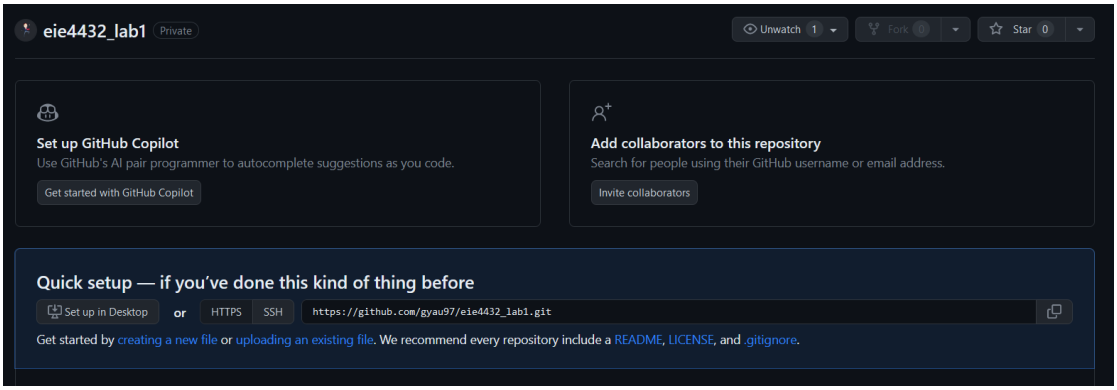
43. Add a new selector to target submit button elements
    - input[type="submit"] {}
44. Set the background color & text color of the submit button using hex code
45. Add 10 pixels of padding vertically and 20 pixels horizontally to the submit button
46. Remove the border around the submit button using "border" property
    - border: https://www.w3schools.com/cssref/pr_border.php

47. Change the cursor style to a pointer when hovering over the submit button using following CSS code
    - cursor: https://www.w3schools.com/cssref/pr_class_cursor.php

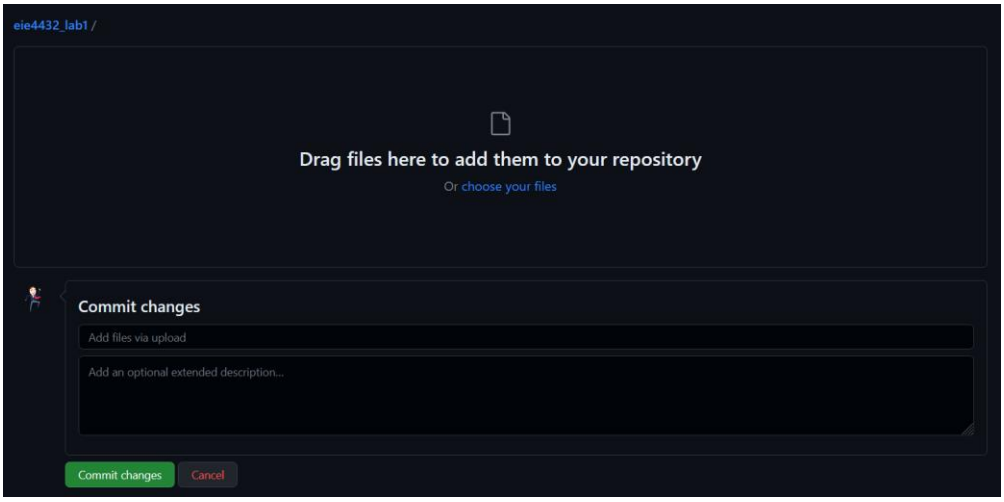
Lab #2.3
Provide a screenshot of the browser screen with your **Student ID and System Date & Time**, which should be showing the whole web form of drink ordering with CSS styles.

Publish your Website (10 mins)

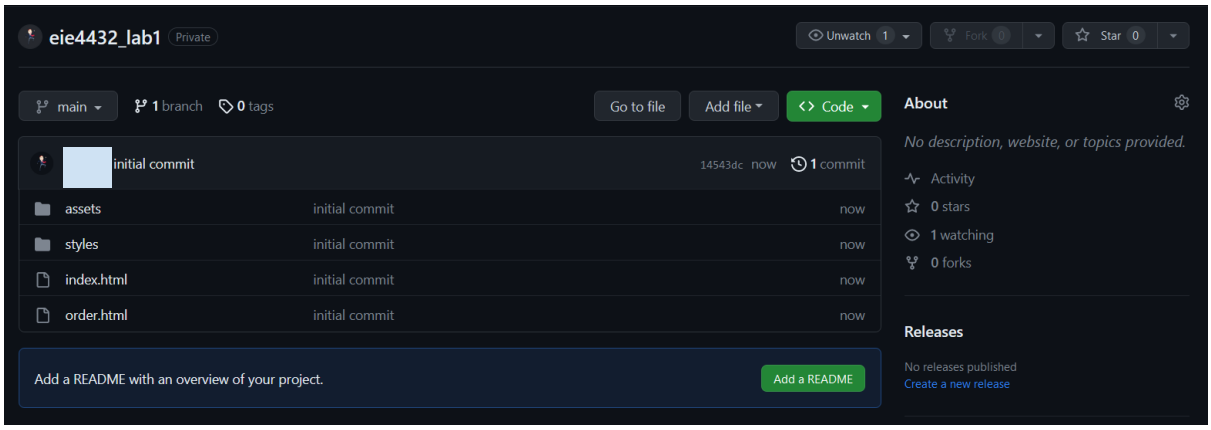
1. Go to GitHub and login your own account
2. Go to “Your Repository” and find “eie4432_lab1” repository that you have created in the beginning of the lab session
3. Click “uploading an existing file”



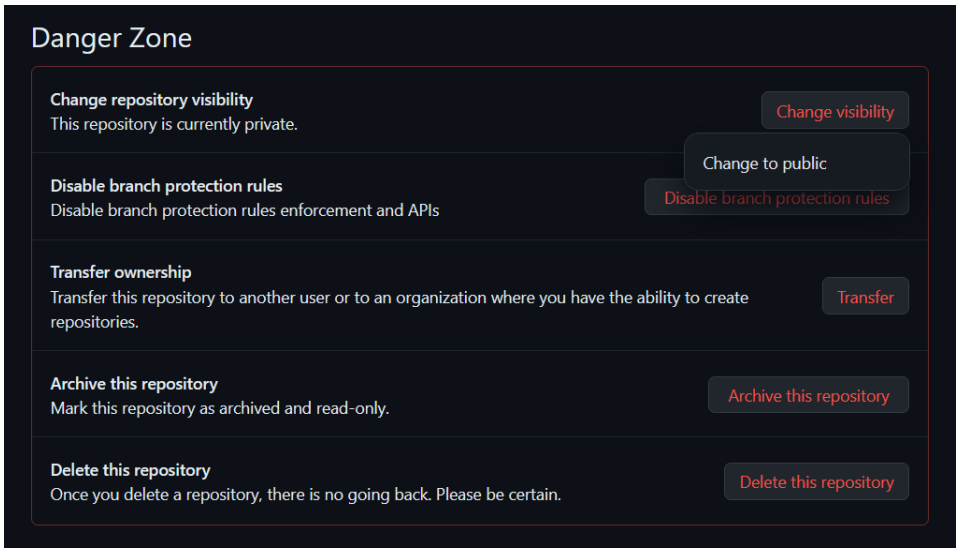
4. Drag the project files into the website, write a commit message “initial commit”, then click “Commit changes” once all files are uploaded succesfully



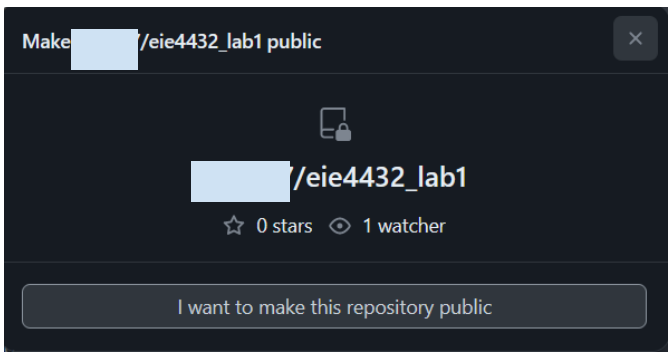
5. Within a minute, you will be redirected to repository page and see all files are uploaded to GitHub successfully



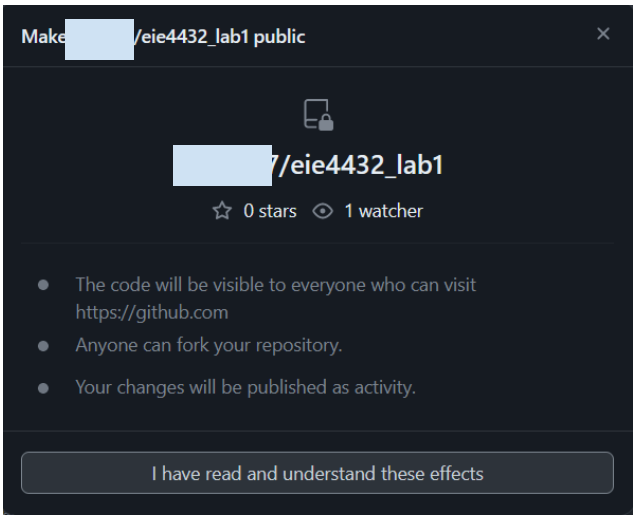
6. To publish the repository to GitHub Pages, you need to make it a public repository first
7. Click “Settings” on top, scroll to “Danger Zone” at the bottom, click “Change visibility” and click “Change to public”



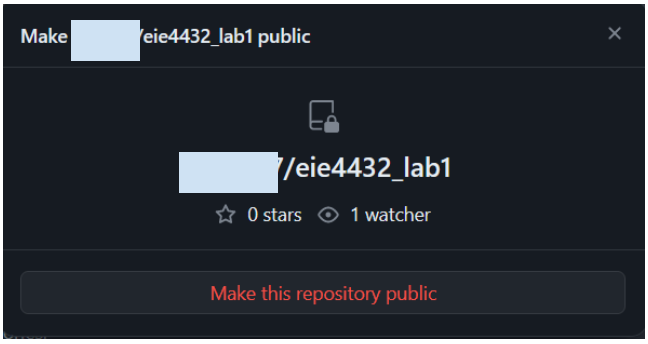
8. Click “I want to make this repository public”



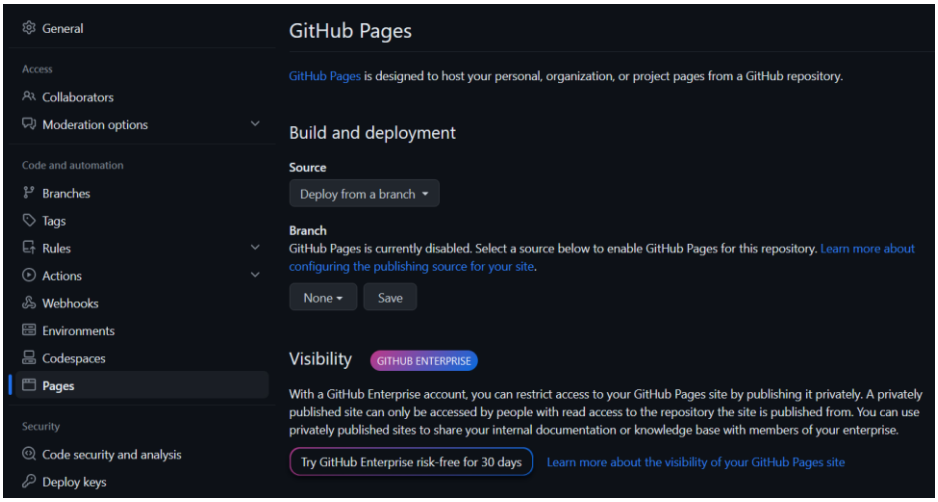
9. Click “I have read and understand these effects”



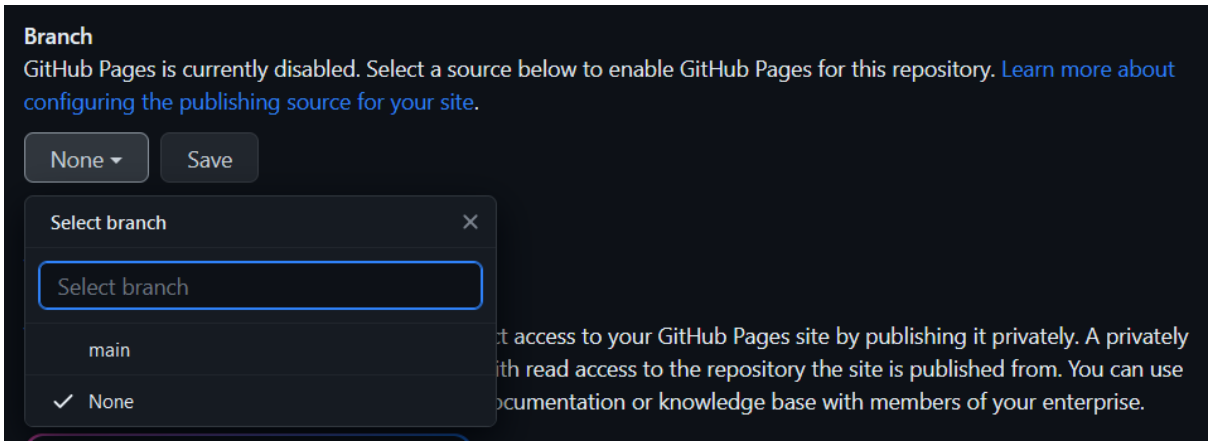
10. Click “Make this repository public”

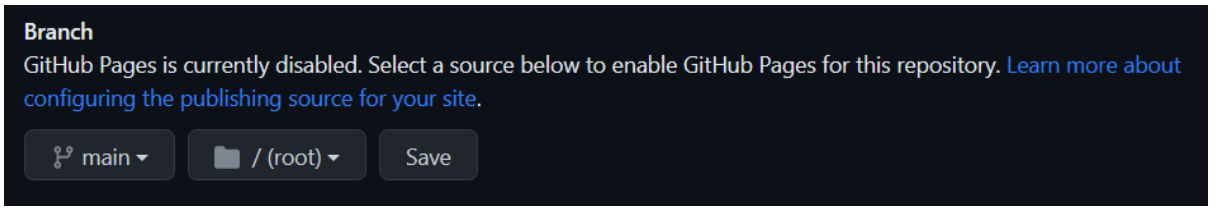


11. You will be asked to type your password to confirm access
12. It is now a public repository, click “Pages” on the left side-bar

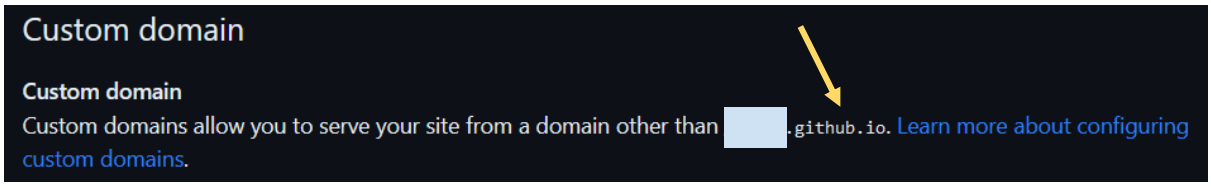


13. Under “Branch” section, choose the branch that you would like to publish
    - It should be “main” branch if you do not create any new branch before
    - Select “main” branch and root folder, click “Save”





14. Find your domain provided by GitHub under “Custom Domain” section



15. Copy the domain name, add “/eie4432_lab1” (the repository name) to browser’s address bar
16. You are now able to share the link to anyone for viewing your website (It takes from 1~3 minutes to be alive)

> **Lab #3.1**
> Provide the public URL of the lab project, eg. https://xxx.github.io/eie4432_lab1/.

END OF LAB 1