//emmet(zen coding)

+ means same level or sibling

> Means child

Example: div#header expands to <div id="header"></div>

Example: div.container expands to <div class="container"></div>

Parent-Child Relationships: Use > to nest elements.

Example: div>ul>li expands to:

<div>

<ul>

<li></li>

</ul>

</div>

Siblings: Use + to add sibling(same alignment) elements.

Example: h1+p expands to:

<h1></h1>

<p></p>

Multiplication: Use \* to repeat elements.

Example: ul>li\*3 expands to:

<ul>

<li></li>

<li></li>

<li></li>

</ul>

Grouping: Use () to group elements.

Example: div>(header>h1)+section>p expands to:

<div>

<header>

<h1></h1>

</header>

<section>

<p></p>

</section>

</div>

Text Content: Use {} to insert text.

Example: p{Hello World} expands to <p>Hello World</p>

//Example

**div.container>p\*2+ul\*5>li\*2**

<div class="container">

    <p></p>

    <p></p>

    <ul>

        <li></li>

        <li></li>

    </ul>

    <ul>

        <li></li>

        <li></li>

    </ul>

    <ul>

        <li></li>

        <li></li>

    </ul>

    <ul>

        <li></li>

        <li></li>

    </ul>

    <ul>

        <li></li>

        <li></li>

    </ul>

</div>

//add full code

html:5

//run

right click->open with live server

//run in browser

view->command pallete->simple browser show->copy and paste address of localhost(http://127.0.0.1:5500/index.html) in vs code->press enter

//css in head

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        body{

            font-family: Arial, Helvetica, sans-serif;

            text-align: center;

        }

        h1{

            color: #FF0000;

        }

    </style>

</head>

<body>

    <h1>Hellow</h1>

    <p>Hi</p>

    <p>Hii</p>

</body>

</html>

//add tailwinds

https://tailwindcss.com/docs/installation/play-cdn

//color

<https://tailwindcss.com/docs/customizing-colors>

//image source

<https://unsplash.com/>

//image in center

<img src="https://images.unsplash.com/photo-1729205940313-ec6f3ccbbc63?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxmZWF0dXJlZC1waG90b3MtZmVlZHw1fHx8ZW58MHx8fHx8"

     alt="Laravel"

     width="500"

     height="500"

     style="display: block; margin: auto;"

     >

//accordion

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="https://cdn.tailwindcss.com"></script>

    <style>

*/\* Styling the accordion \*/*

        details {

            border: 1px solid #ccc;

            border-radius: 5px;

            margin: 5px 0;

            padding: 5px;

            max-width: 100%;

        }

        summary {

            font-weight: bold;

            cursor: pointer;

            padding: 5px;

        }

        details[open] summary {

            color: #007BFF;

*/\* Color when section is expanded \*/*

        }

*/\* Optional: style the content \*/*

        p {

            padding: 10px;

            margin: 0;

            background-color: #f9f9f9;

            border-top: 1px solid #ccc;

        }

    </style>

</head>

<body>

    <div class="accordion">

        <details>

            <summary>Section 1</summary>

            <p>This is the content of Section 1. It becomes visible when you expand the section.</p>

        </details>

        <details>

            <summary>Section 2</summary>

            <p>This is the content of Section 2. It becomes visible when you expand the section.</p>

        </details>

        <details>

            <summary>Section 3</summary>

            <p>This is the content of Section 3. It becomes visible when you expand the section.</p>

        </details>

    </div>

</body>

**//extension**

//theme

one dark pro

//formatting document

prettier

view->command palette->format document

//php

devsense

//javascript hint

eslint

//GitHub code hint(search with only github)

GitHub Copilot

//Arrange Assets folder

//arrange assets folder

* assets
  + - css
      * + style.css
    - images
      * + image files
    - js
      * + demo.js

//Link with css

//css code writtend assets/css/style.css file

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="https://cdn.tailwindcss.com"></script>

    <link rel="stylesheet" href="assets/css/style.css">

</head>

<body>

    <div class="accordion">

        <details>

            <summary>Section 1</summary>

            <p>This is the content of Section 1. It becomes visible when you expand the section.</p>

        </details>

        <details>

            <summary>Section 2</summary>

            <p>This is the content of Section 2. It becomes visible when you expand the section.</p>

        </details>

        <details>

            <summary>Section 3</summary>

            <p>This is the content of Section 3. It becomes visible when you expand the section.</p>

        </details>

    </div>

</body>

</html>

//Add javascript

<body>

</body>

<script src="assets/js/script.js"></script>

</html>

//in assets/js folder

alert("Hellow from utsab");

//Attractive FORM

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Styled Registration Form</title>

    <style>

*/\* Basic form container styling \*/*

        .form-container {

            max-width: 500px;

            margin: 0 auto;

            padding: 20px;

            border: 1px solid #ccc;

            border-radius: 10px;

            background-color: #f9f9f9;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        h2 {

            text-align: center;

            color: #333;

        }

        label {

            display: block;

            margin: 10px 0 5px;

            font-weight: bold;

            color: #555;

        }

        input[type="text"],

        input[type="email"],

        input[type="password"],

        select {

            width: 100%;

            padding: 10px;

            margin-bottom: 15px;

            border: 1px solid #ccc;

            border-radius: 5px;

            font-size: 16px;

            box-sizing: border-box;

        }

*/\* Radio buttons and checkboxes inline \*/*

        .radio-group,

        .checkbox-group {

            display: flex;

            align-items: center;

            gap: 10px; */\* space between options \*/*

        }

        input[type="submit"] {

            width: 100%;

            padding: 10px;

            font-size: 16px;

            color: #fff;

            background-color: #007BFF;

            border: none;

            border-radius: 5px;

            cursor: pointer;

            transition: background-color 0.3s ease;

        }

        input[type="submit"]:hover {

            background-color: #0056b3;

        }

    </style>

</head>

<body>

<div class="form-container">

    <h2>Registration Form</h2>

    <form action="submit\_form.php" method="post">

        <label for="name">Full Name:</label>

        <input type="text" id="name" name="name" required>

        <label for="email">Email:</label>

        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" required>

        <label>Gender:</label>

        <div class="radio-group">

            <input type="radio" id="male" name="gender" value="male">

            <label for="male">Male</label>

            <input type="radio" id="female" name="gender" value="female">

            <label for="female">Female</label>

        </div>

        <label for="country">Country:</label>

        <select id="country" name="country" required>

            <option value="">Select your country</option>

            <option value="usa">United States</option>

            <option value="canada">Canada</option>

            <option value="uk">United Kingdom</option>

        </select>

        <label>Interests:</label>

        <div class="checkbox-group">

            <input type="checkbox" id="sports" name="interests" value="sports">

            <label for="sports">Sports</label>

            <input type="checkbox" id="music" name="interests" value="music">

            <label for="music">Music</label>

            <input type="checkbox" id="reading" name="interests" value="reading">

            <label for="reading">Reading</label>

        </div>

        <input type="submit" value="Submit">

    </form>

</div>

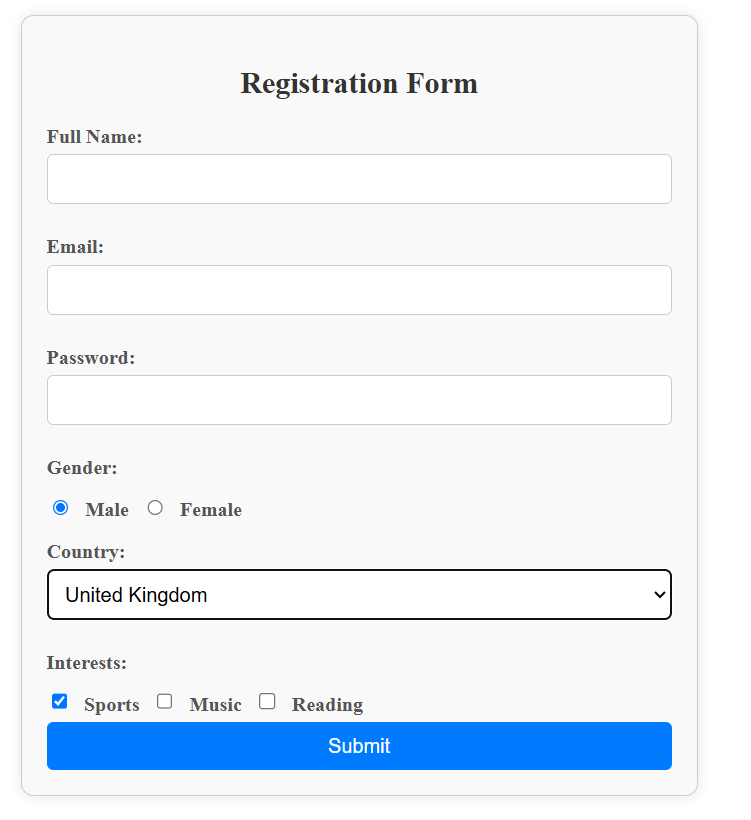
</body>

</html>

//Git and Github

Developers use Git on their local machine to manage code versions.

When they’re ready to share the code or back it up, they push it to a remote repository on GitHub.



**//Explanation**

**HTML Structure**

**1. Container for the Form**

<div class="form-container">

The form is wrapped in a <div> with the class form-container, which allows us to style the form separately from the rest of the page. This container adds padding, borders, background color, and a shadow effect.

**2. Form Header**

<h2>Registration Form</h2>

This <h2> element provides a title for the form, centered at the top.

**3. Form Element**

<form action="submit\_form.php" method="post">

The <form> tag defines the start of the form:

* action="submit\_form.php" specifies where the form data will be sent when the form is submitted. Here, it goes to a file called submit\_form.php.
* method="post" sends the form data securely without displaying it in the URL, as opposed to get.

**Input Fields**

**Full Name, Email, and Password**

Each input for the name, email, and password is created with <input type="text">, <input type="email">, and <input type="password">. The required attribute ensures that users can’t submit the form without filling these fields.

**Gender (Radio Buttons)**

<div class="radio-group">

<input type="radio" id="male" name="gender" value="male">

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="female">

<label for="female">Female</label>

</div>

The radio-group div groups the radio buttons. Using display: flex; on this group aligns them in a single line.

Each radio button:

* Has type="radio".
* Has the name="gender" attribute, meaning only one option can be selected.
* Has an id that matches the corresponding <label for=""> element, allowing users to click the label to select the button.

**Country (Dropdown)**

<label for="country">Country:</label>

<select id="country" name="country" required>

<option value="">Select your country</option>

<option value="usa">United States</option>

<option value="canada">Canada</option>

<option value="uk">United Kingdom</option>

</select>

The <select> tag creates a dropdown list with various <option> tags, each representing a selectable country. The required attribute ensures the user must choose a country before submission.

**Interests (Checkboxes)**

<div class="checkbox-group">

<input type="checkbox" id="sports" name="interests" value="sports">

<label for="sports">Sports</label>

<input type="checkbox" id="music" name="interests" value="music">

<label for="music">Music</label>

<input type="checkbox" id="reading" name="interests" value="reading">

<label for="reading">Reading</label>

</div>

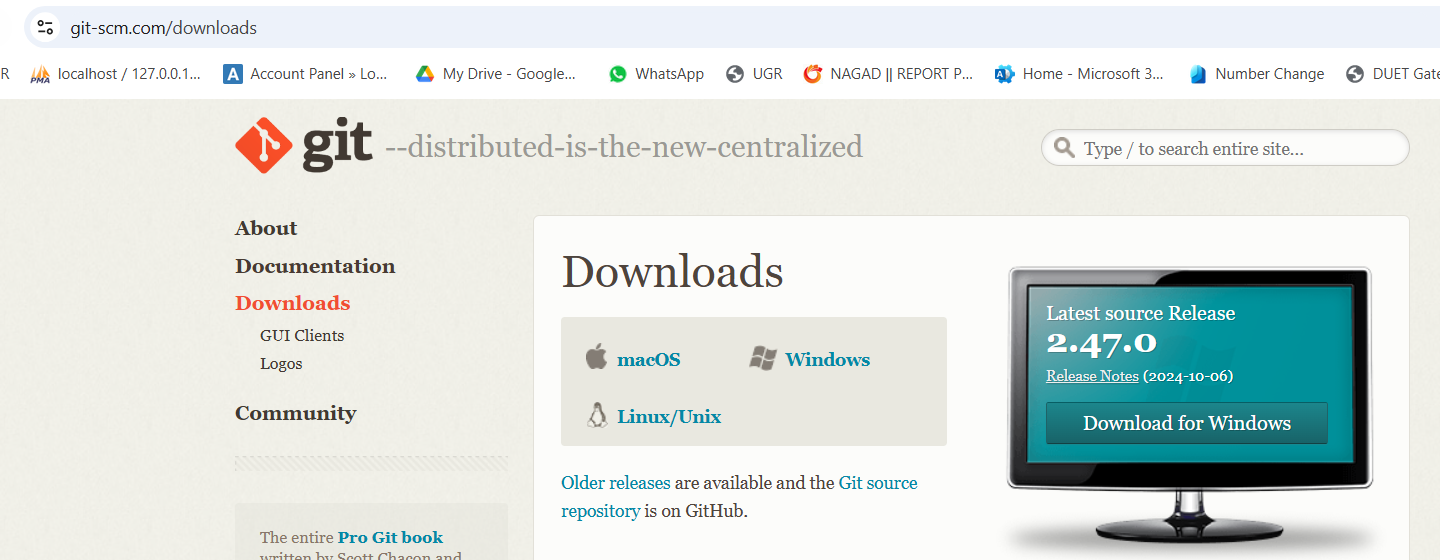
The checkbox-group div groups the checkboxes, so they display in a single line. Users can select multiple checkboxes at once since they are each independent.

**Submit Button**

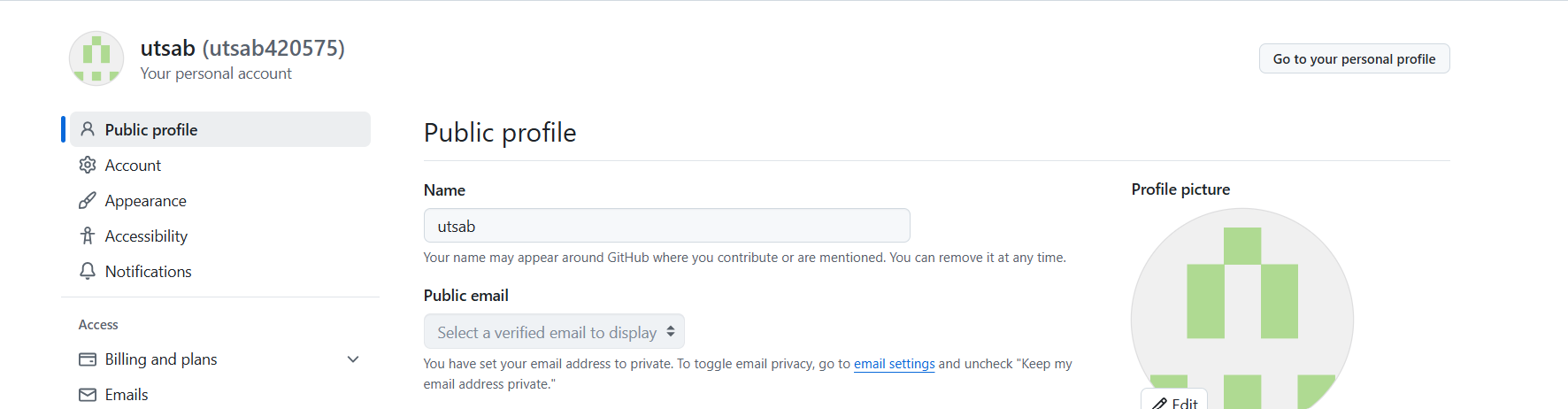
<input type="submit" value="Submit">

This button submits the form. The value="Submit" text appears on the button.

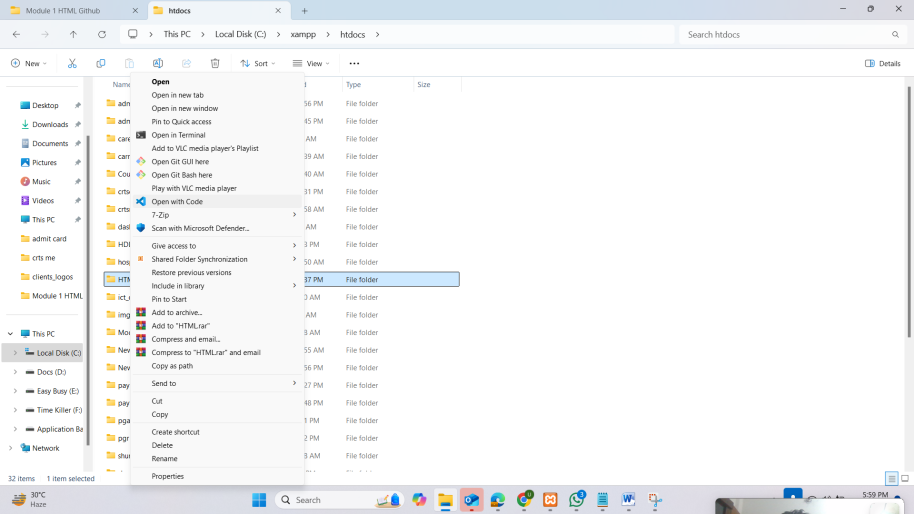
//1. Download Git



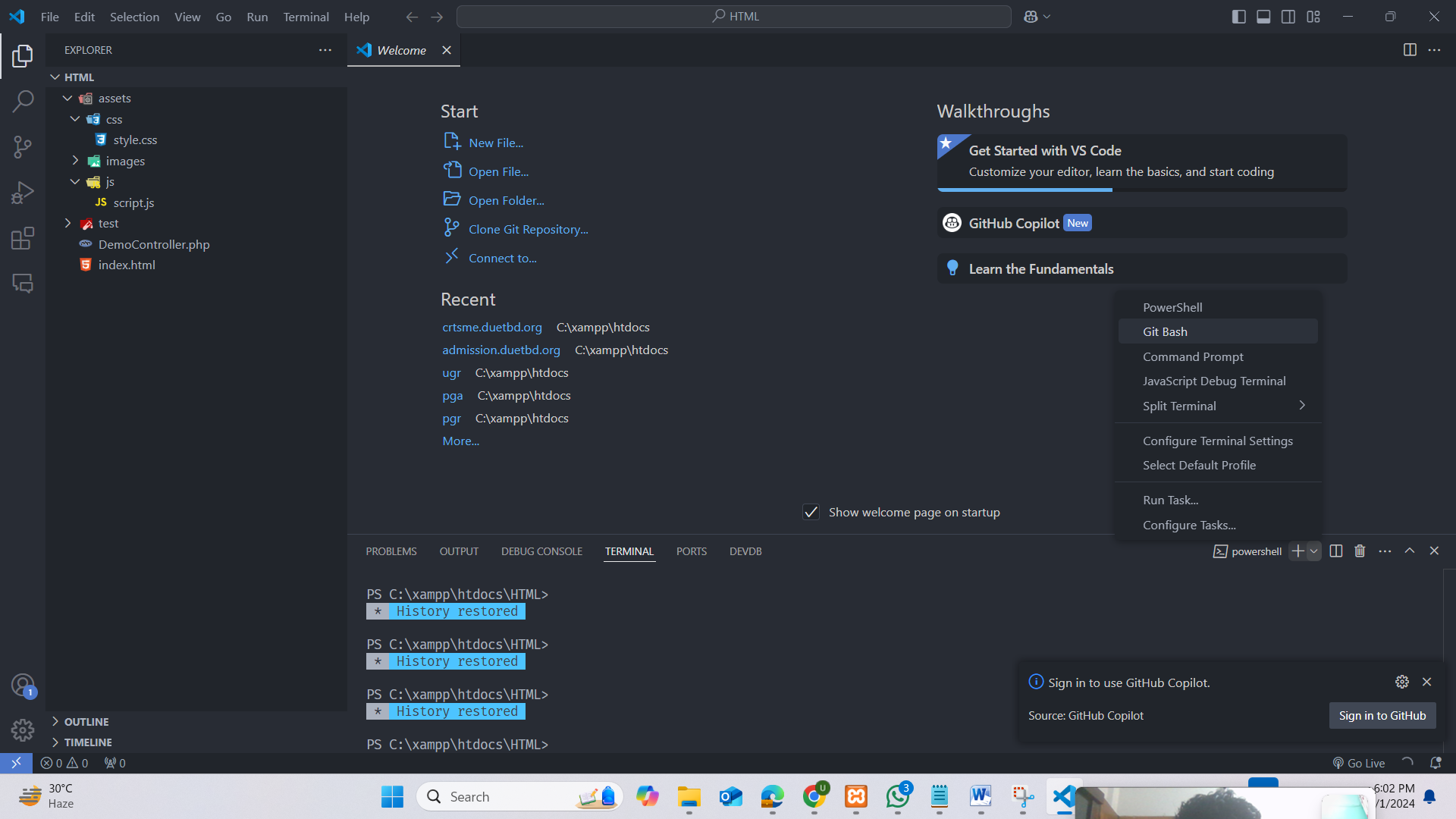
//2. Going to github and see email address and name



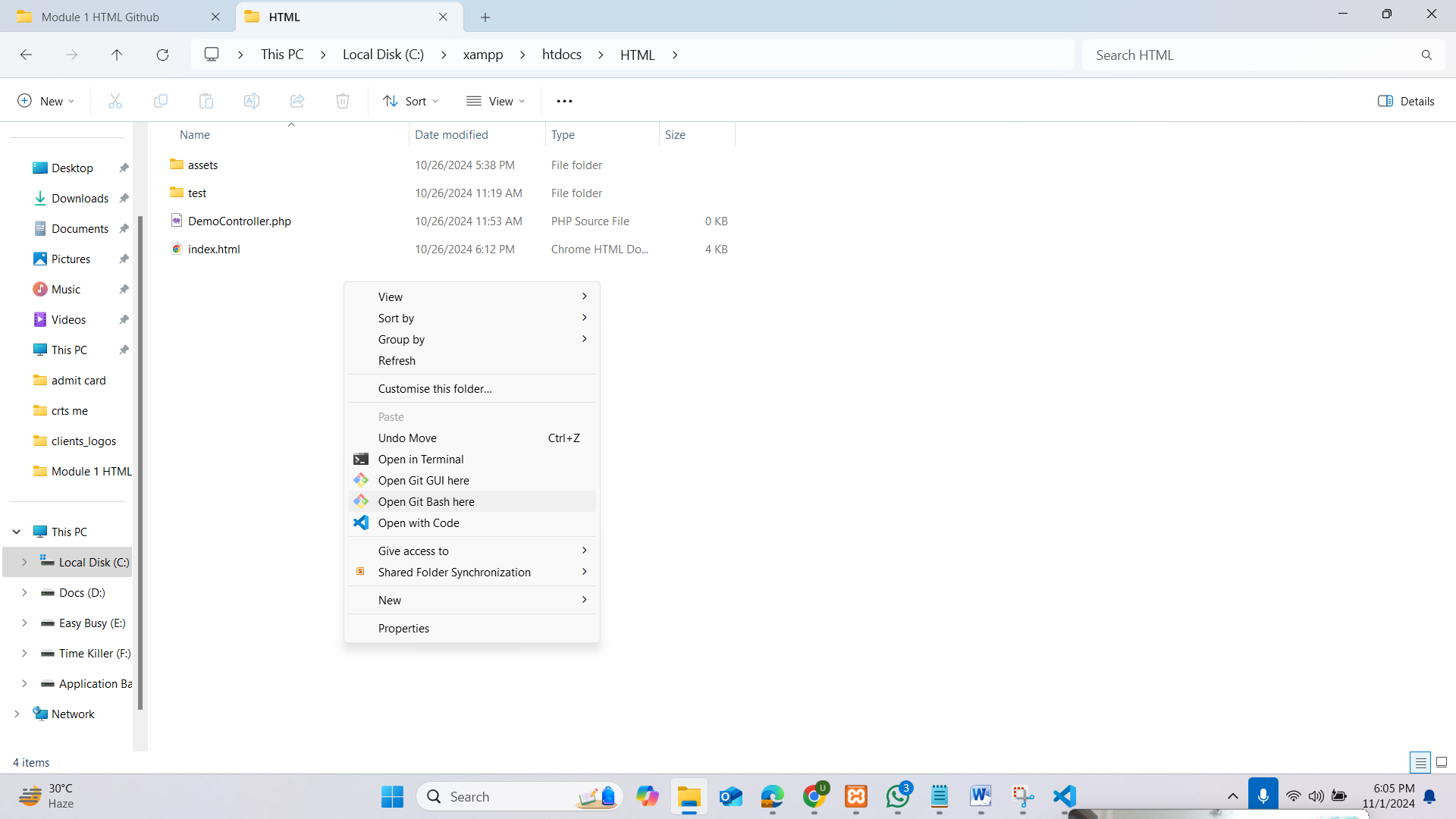
//3.Open Project folder into vs code

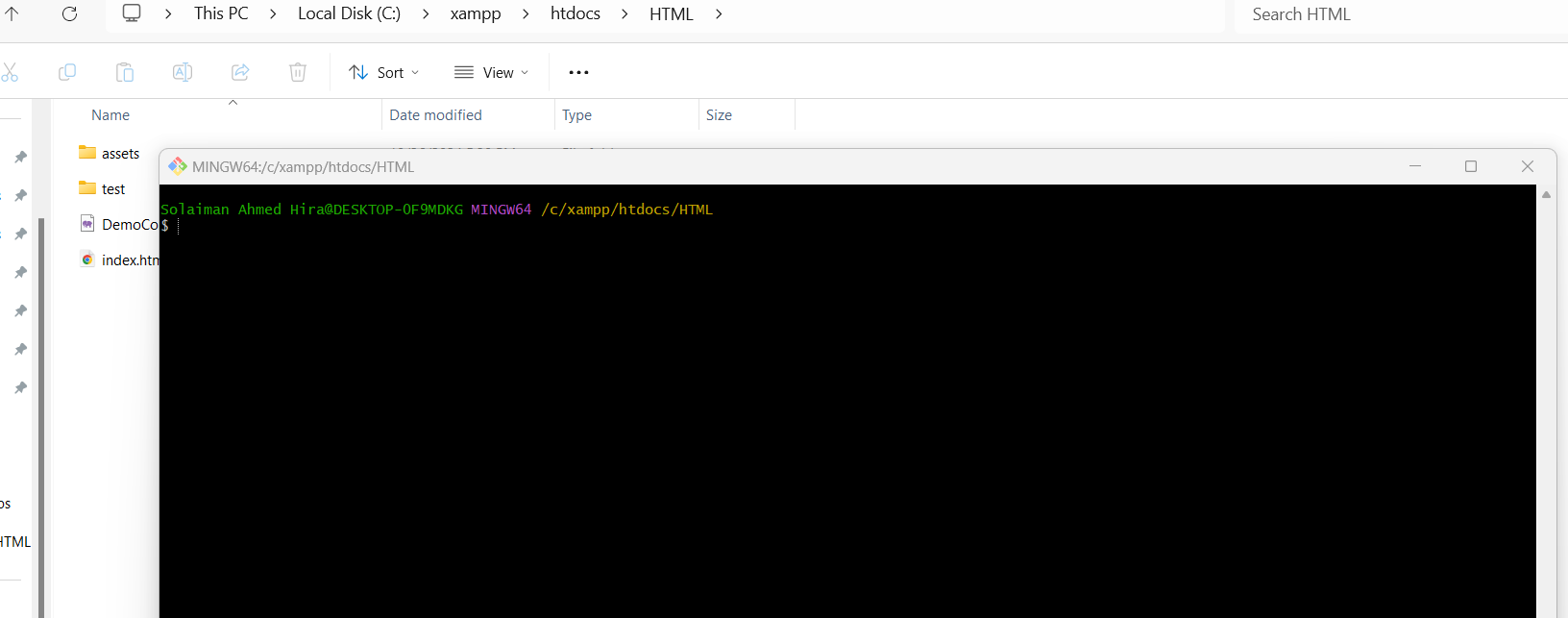


4.Open terminal in gitbash



4.Or Right Click and Open git bash here

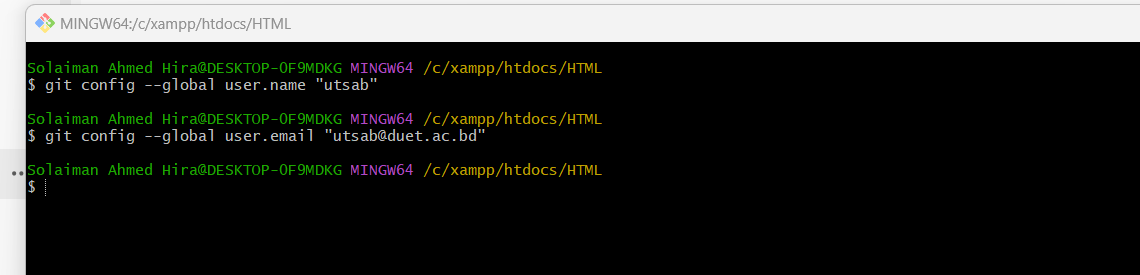




5. Set up your Git identity, which will be used for all commits(user name and email address of github)

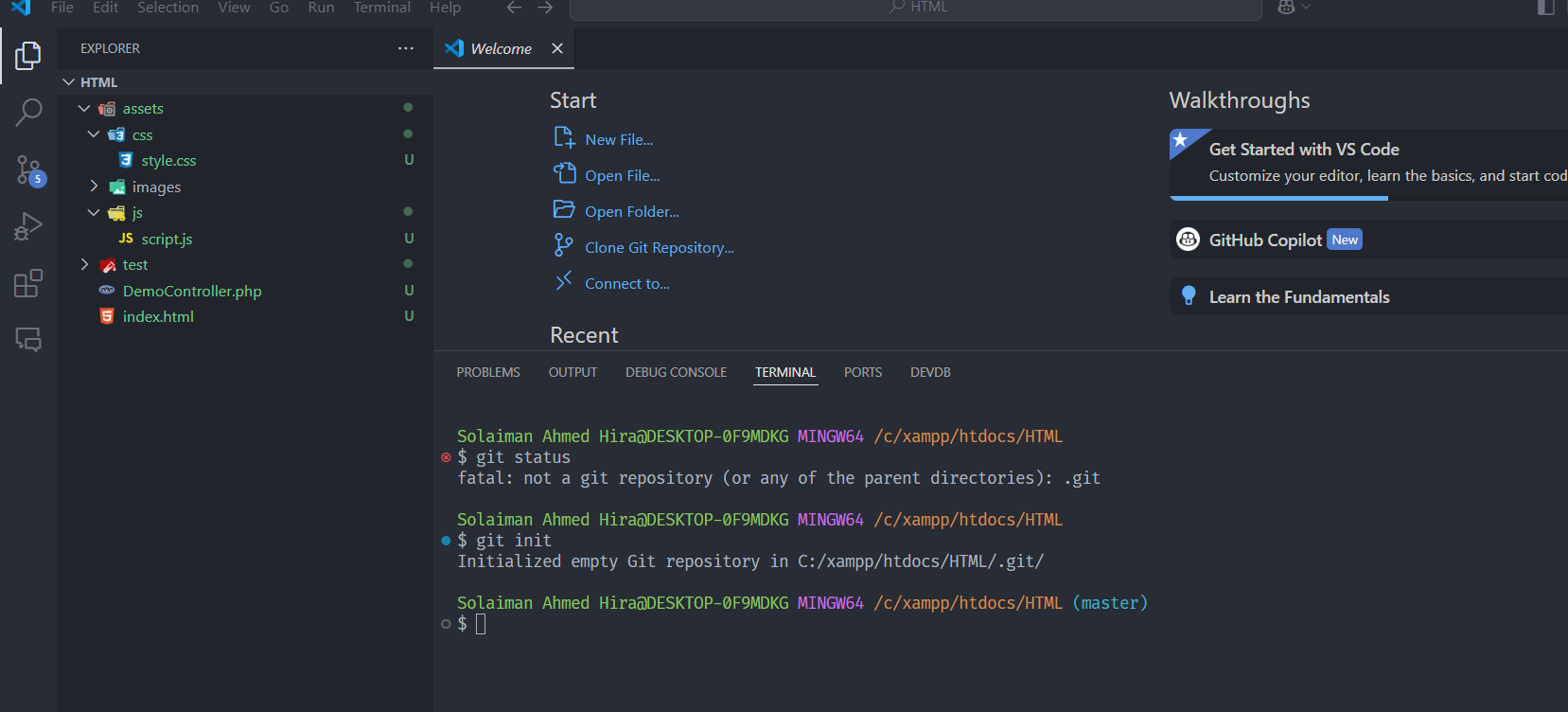
**$ git config --global user.name "utsab"**

**$ git config --global user.email "utsab@duet.ac.bd"**



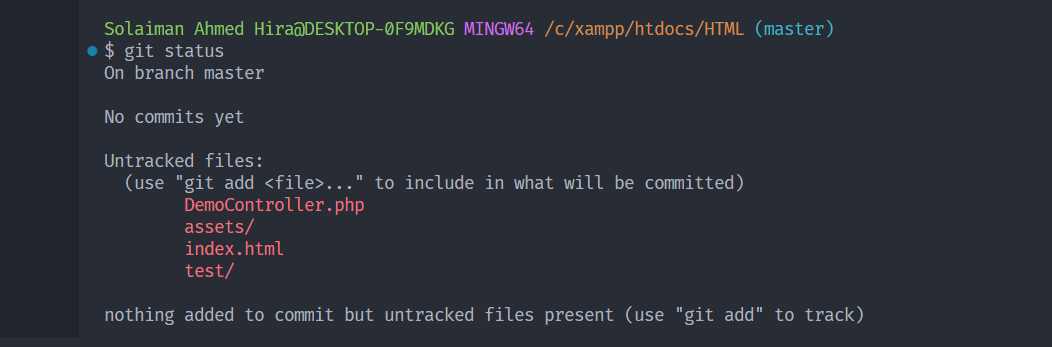
6. Create a New Git Repository , **Initialize a new repository**: Navigate to your project folder and run:

**git init**



7.Here we see which file in untracked or not committed.

**git status**



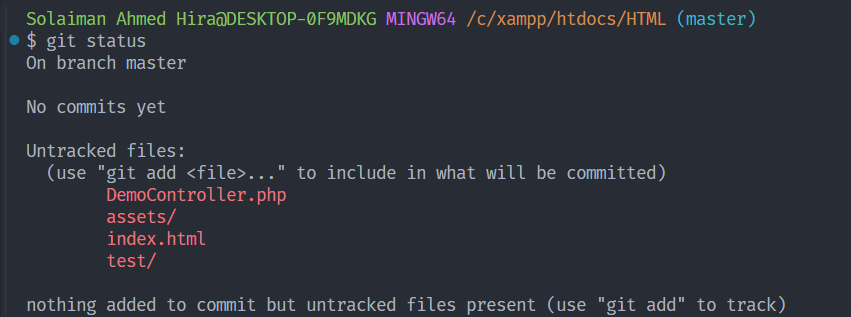
7. **Stage changes**: Add files to the staging area before committing:

//Add single file

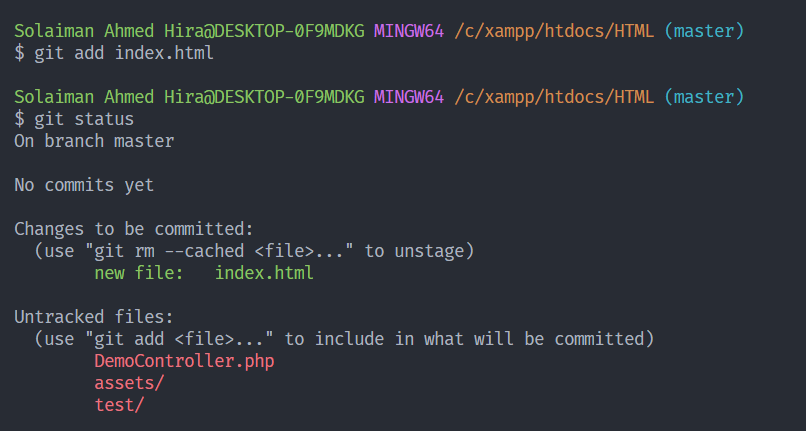
**git add filename**

//Add all file

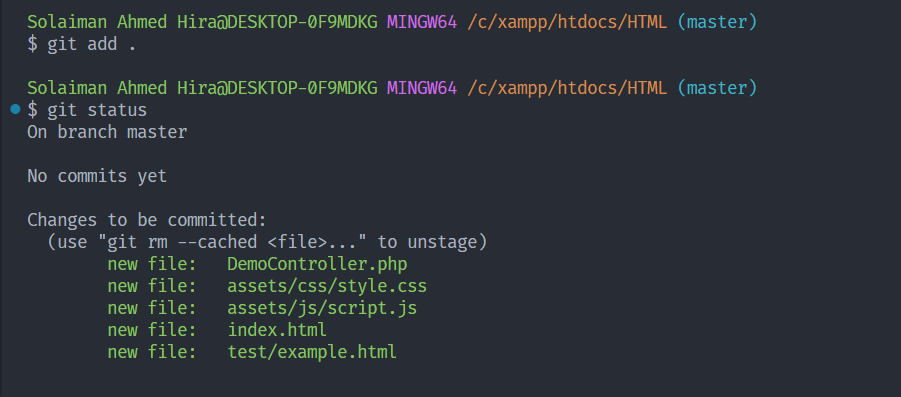
**git add .**



//Add single file

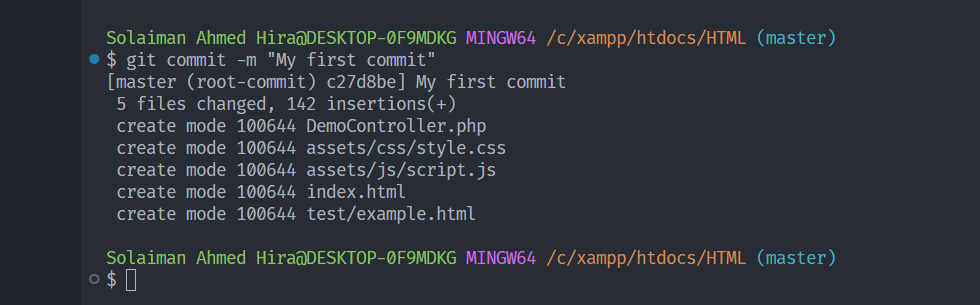


//Add all file



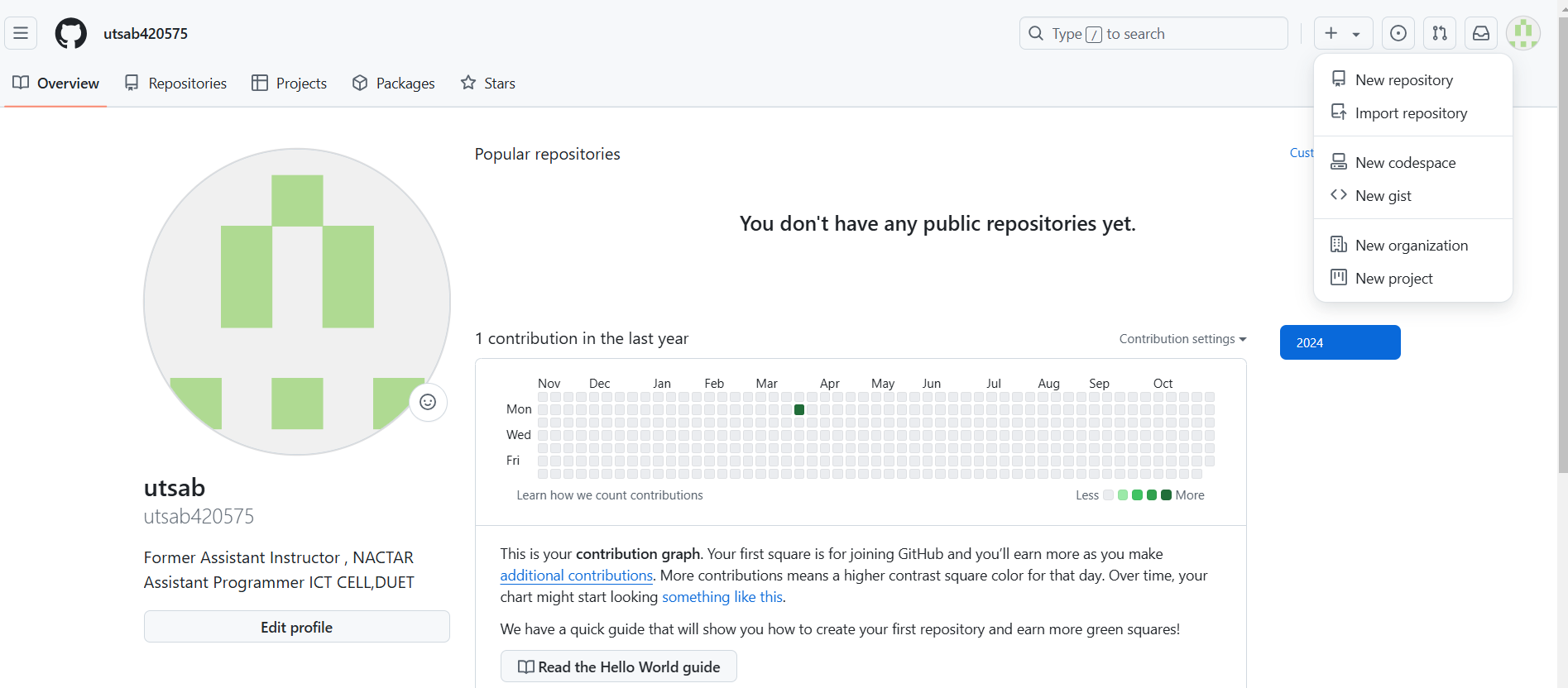
8. **Commit changes**: Commit staged changes with a descriptive message.

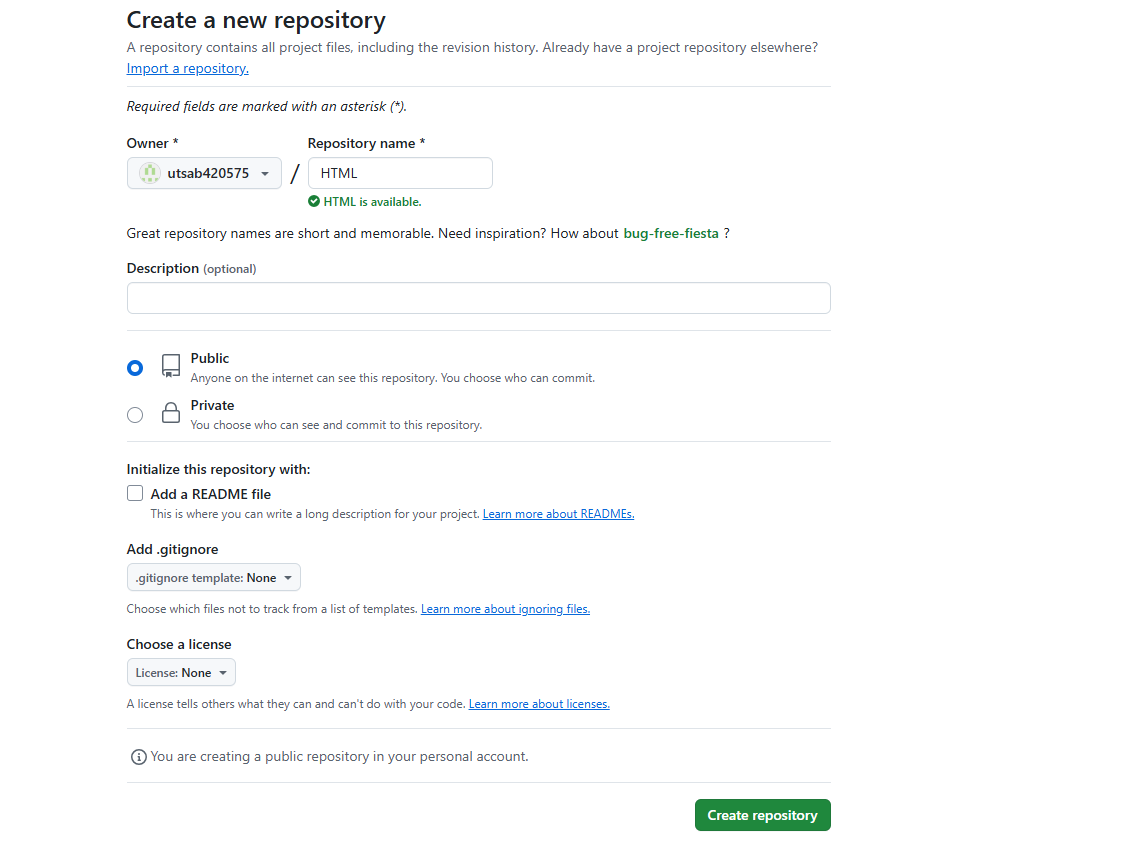
**git commit -m "My first commit"**



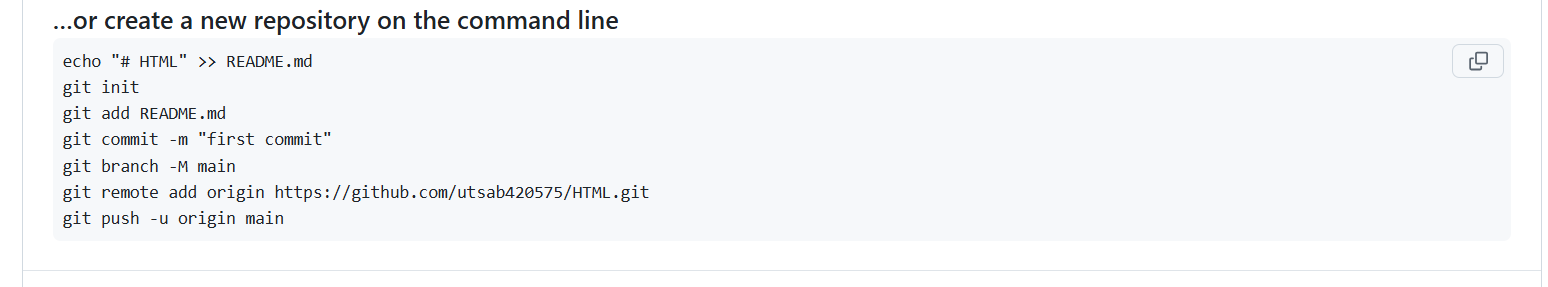
//9. Now we push this into GitHub

Create New Repository

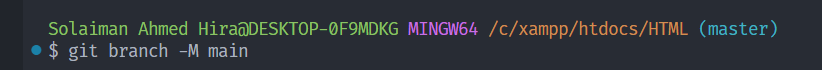




10.Change Master to Main(It’s for first time only)



**git branch -M main**

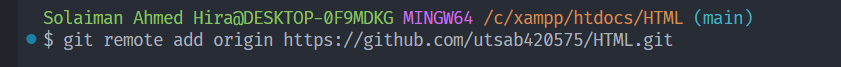


11. **Link the Local Repository to GitHub.**

git remote add origin https://github.com/your-username/your-repo.git

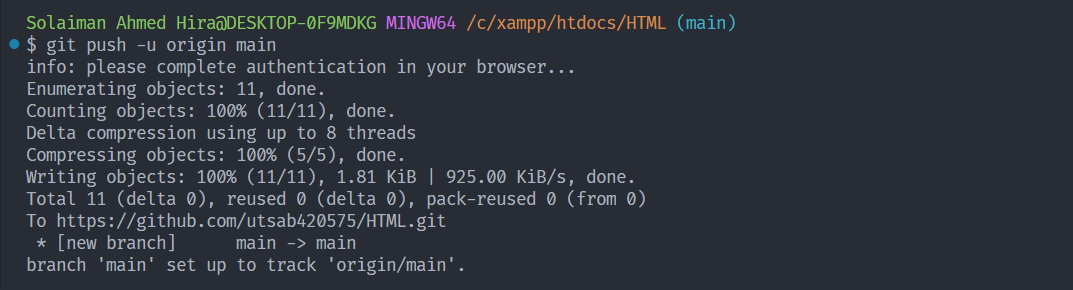
**git remote add origin https://github.com/utsab420575/HTML.git**

In your local project folder, link the local repository to the GitHub repository you just created by adding it as a remote:



12.If this is the first time you’re pushing, use:

**git push -u origin main**



Next time just push

**git push**

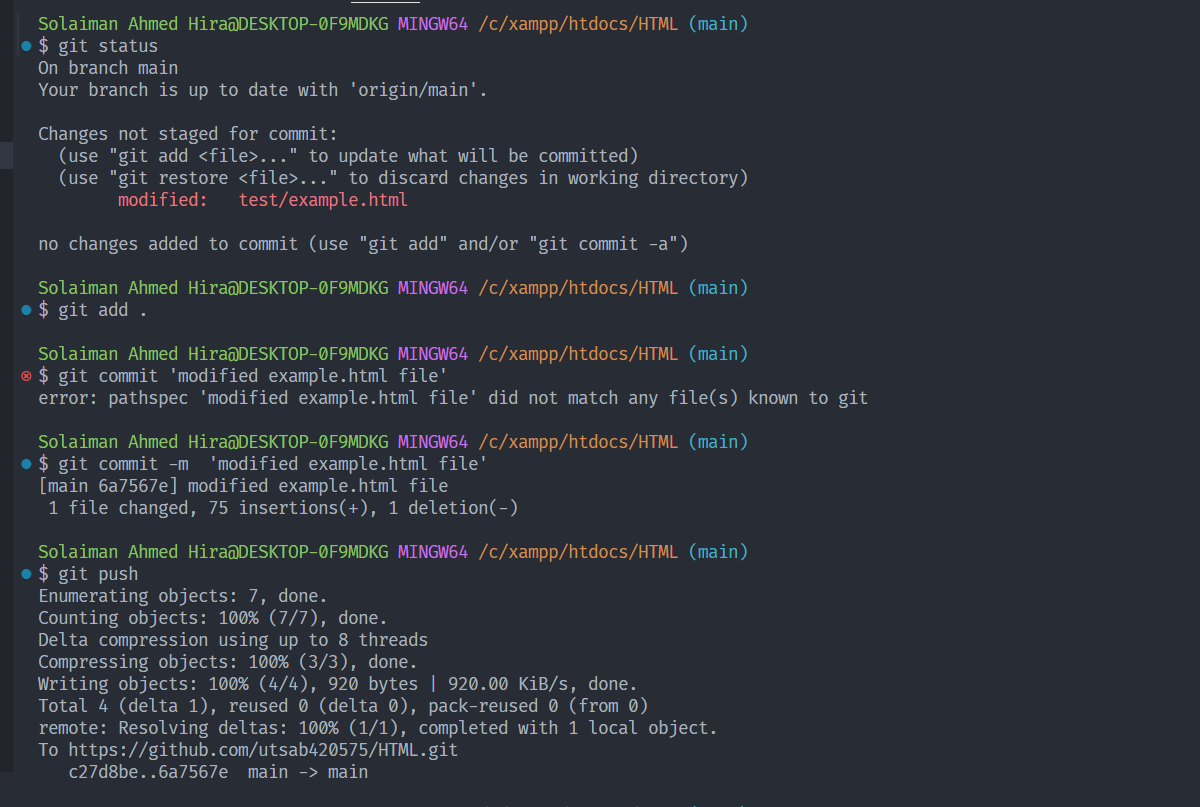
1.After Adding some html file or modified file we repeat above steps;

**git status**

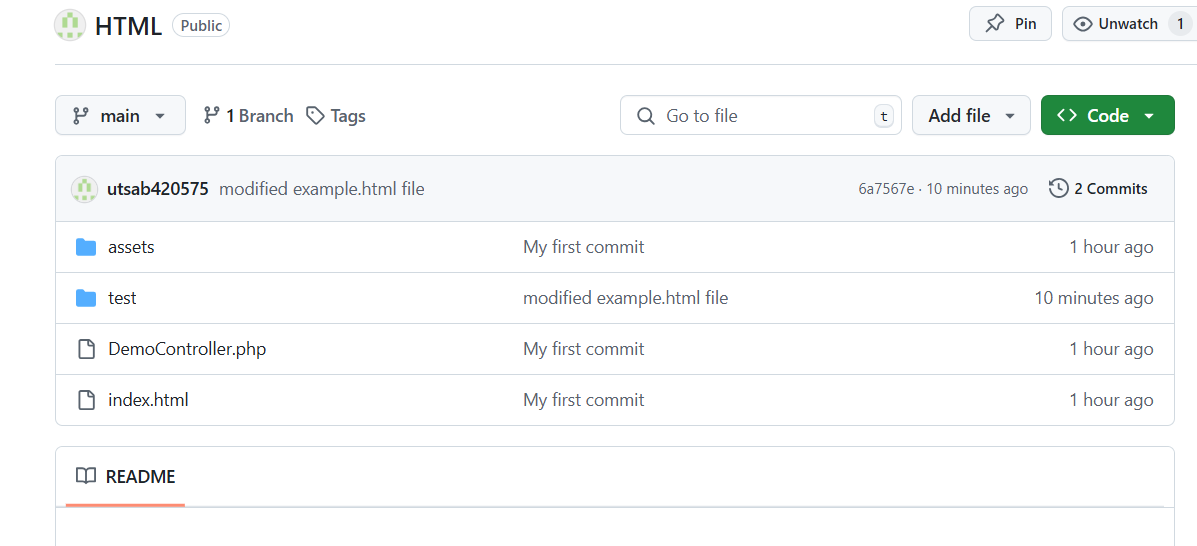
**git add .**

**git commit -m "My first commit"**

**git push**

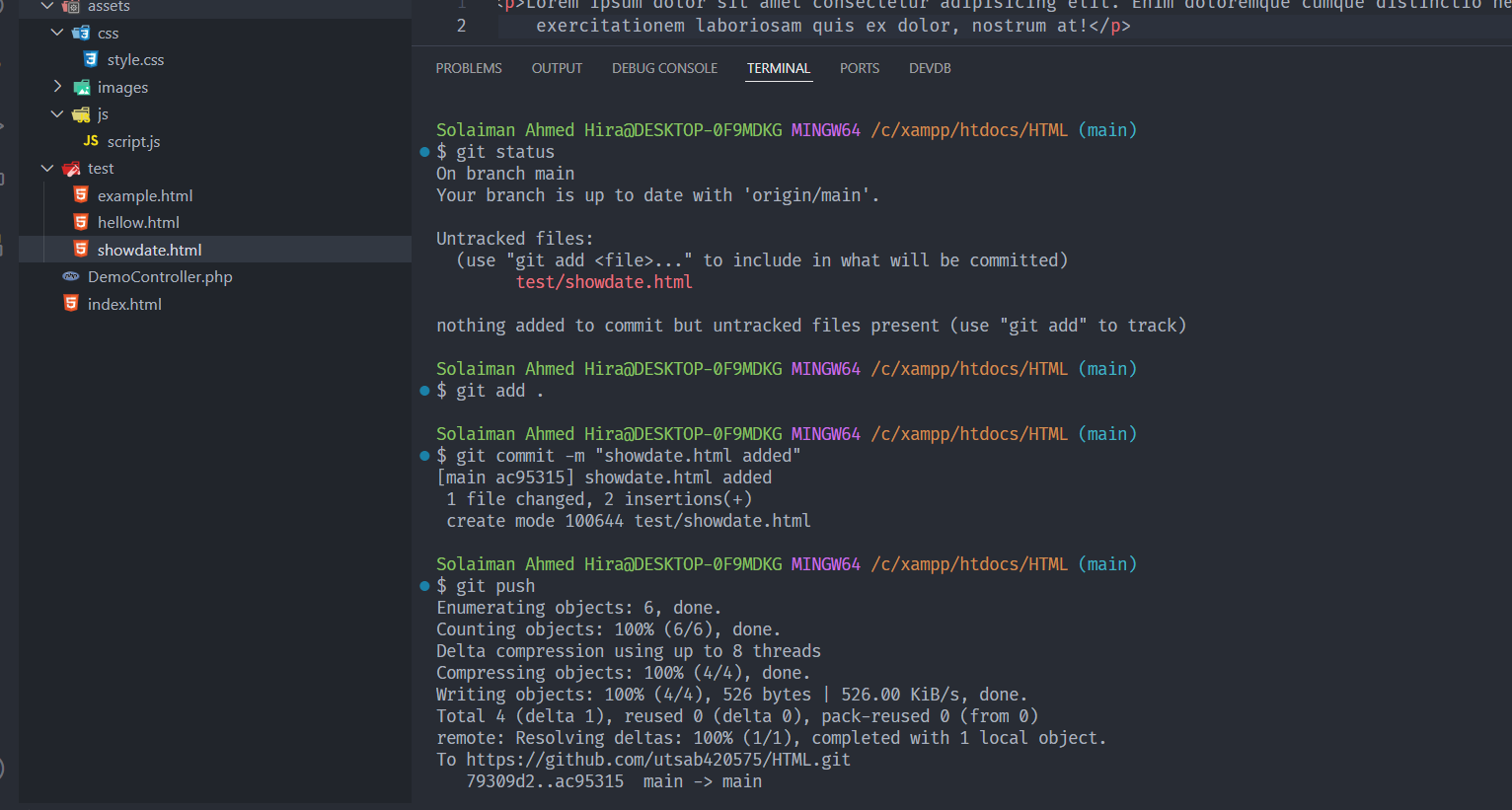


After push all file newly stored into github



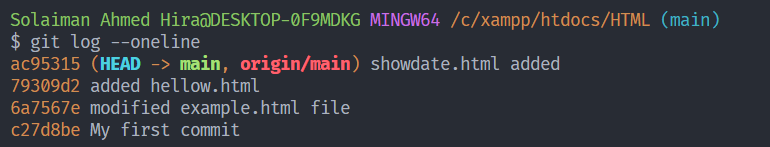
Git Rollback/Revert (Here we rollback to specific commit)

1.We added some file



2. search commit has

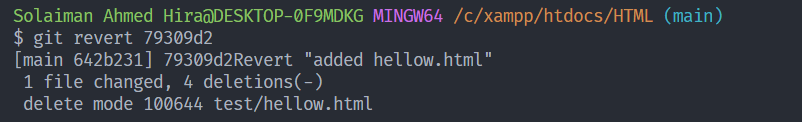
git log –oneline



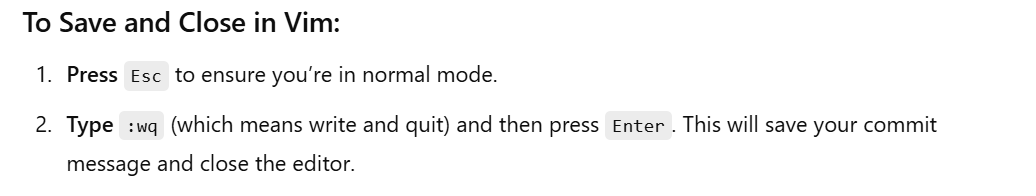
3. Identify the Commit to Revert:

4. Use the git revert Command:

git revert 79309d2

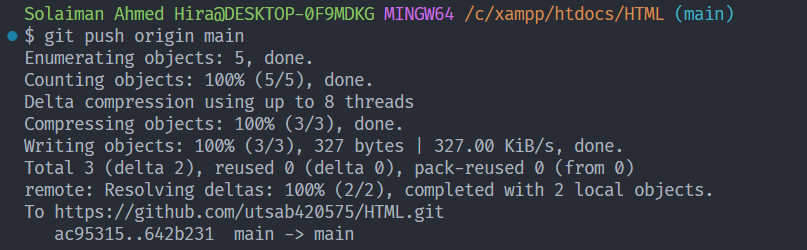


5.Press save and close if open in vs code  
otherwise use this bellow command   
press esc and :wq

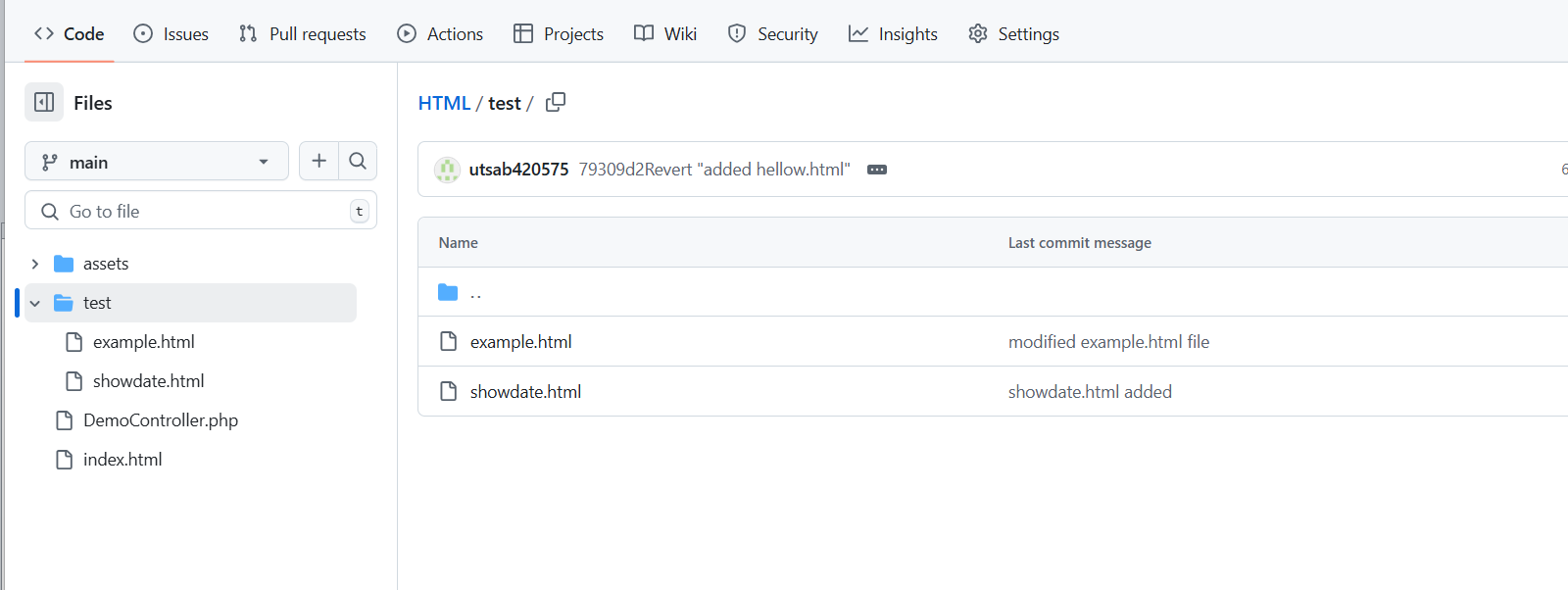


6.Finally push into github

git push origin main

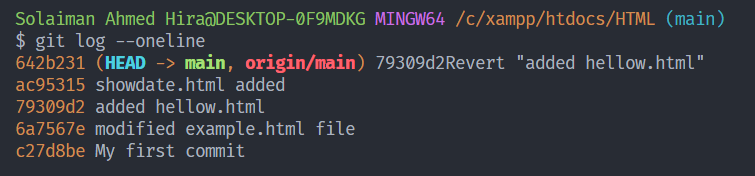


hellow.html file is deleted and all other things not deleted from github

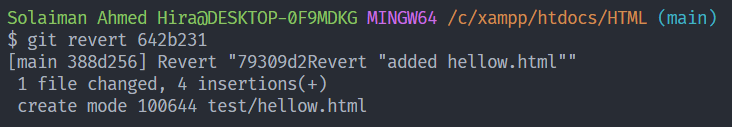


Git Restoring / File Rollback

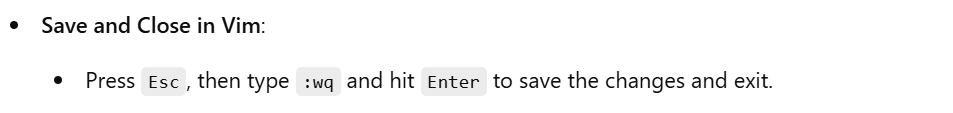
1.git log --oneline



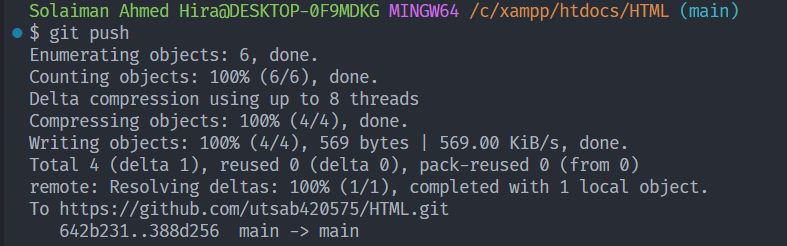
**2.** git revert 642b231

****

**3.** Complete the Commit Message

****

**4.** Push the Changes to GitHub.

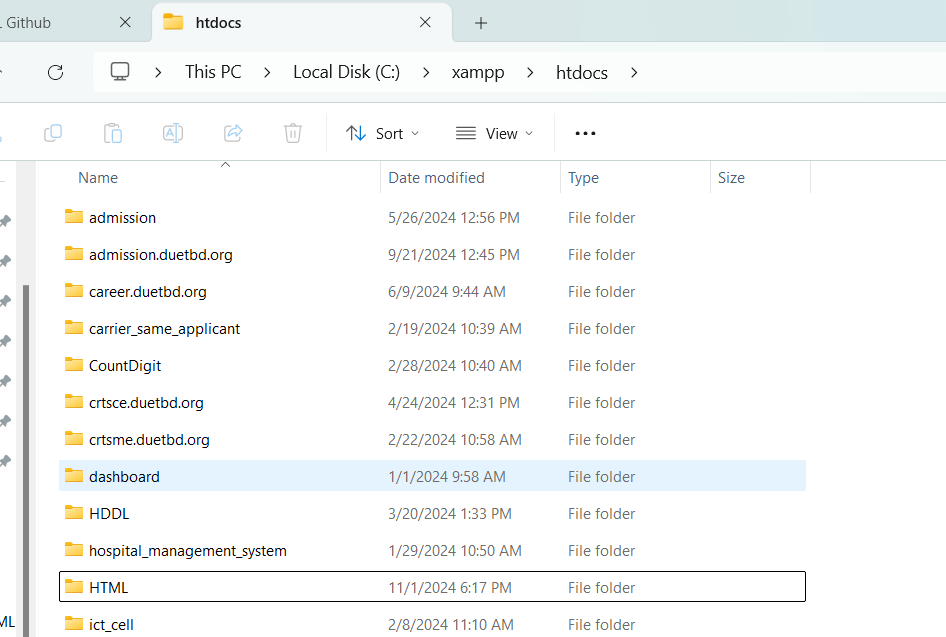
****

**Visual Studio As A Default Browser**

****

**Now how to clone file**

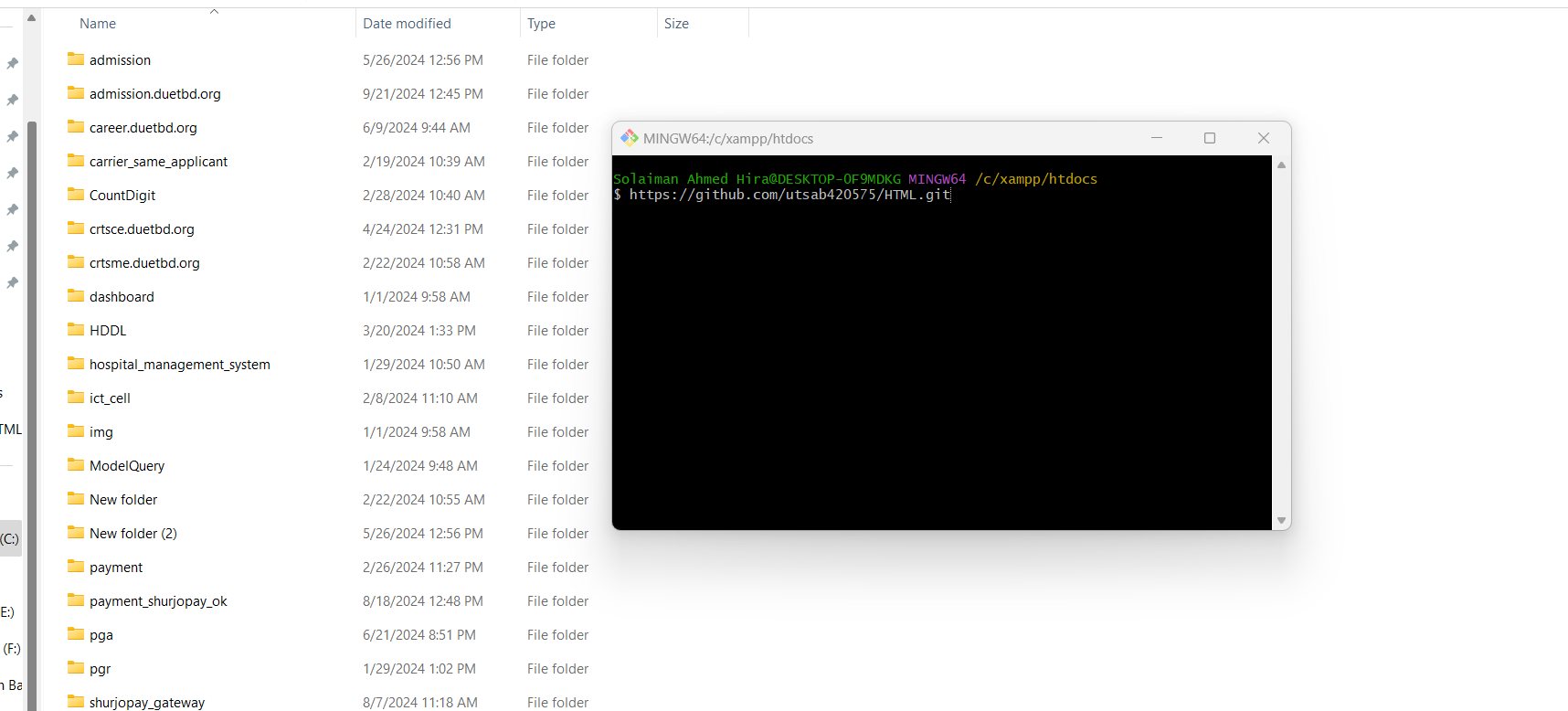
**1. I deleted HTML folder(Or I am to going to work with another PC)**

****

2.Go to Github

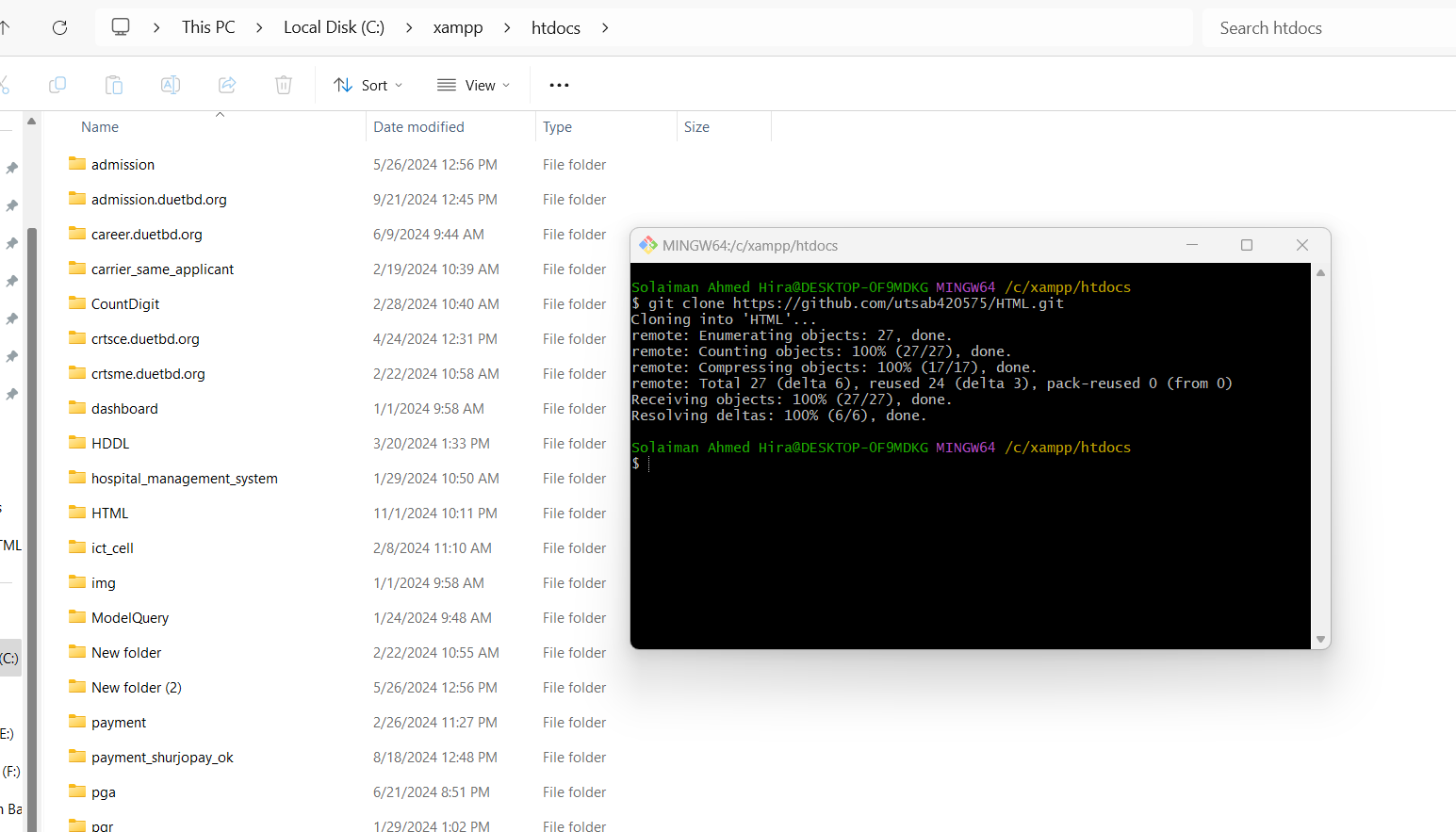


3.Open Git Bash Where Folder will be placed

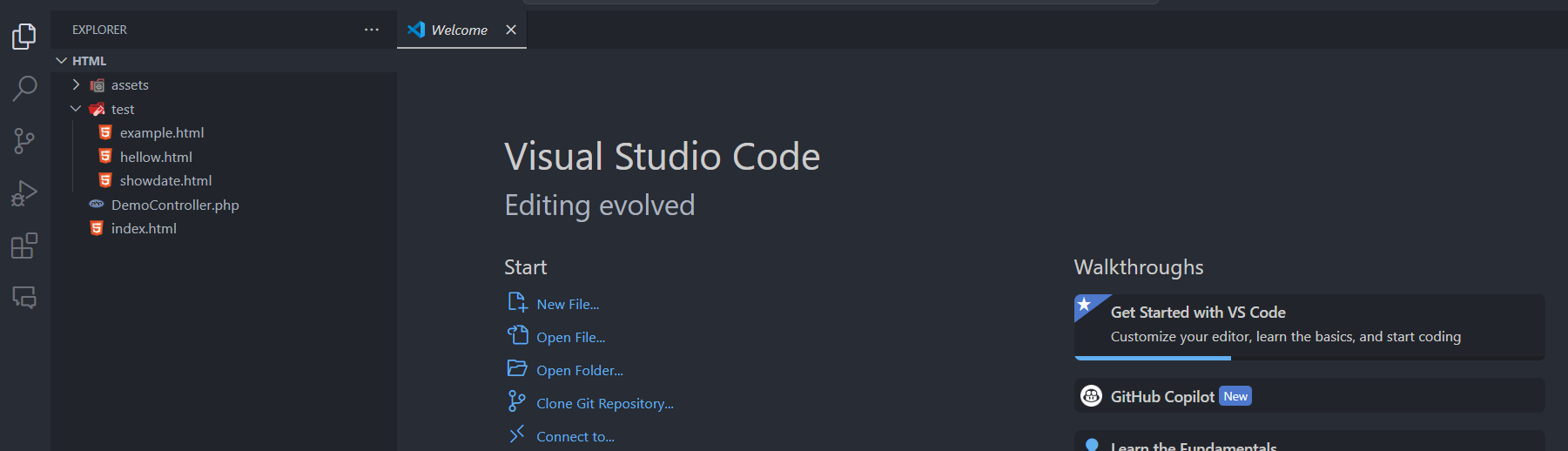


3.press git clone command

git clone https://github.com/utsab420575/HTML.git



4.Open in vs code



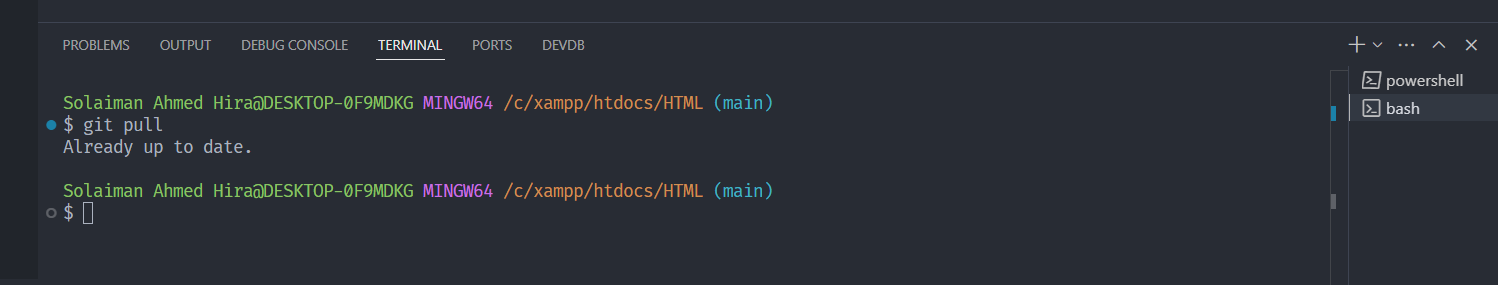
Git PULL

1. If we work in another pc(Office PC) and push into github

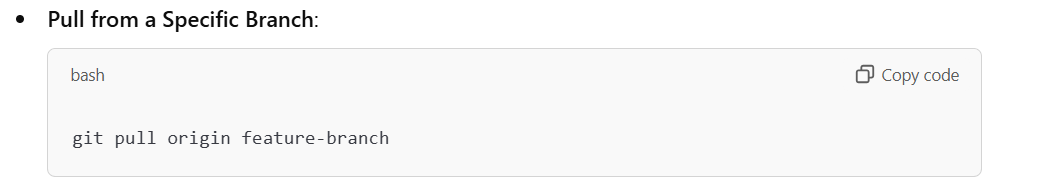
Now I come to home and pull the updated code into my home pc VS CODE



Or, This pulls from the default remote branch (usually origin/main).



Pull from specific branch:



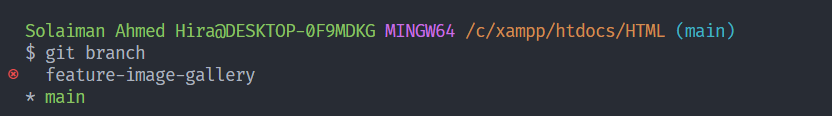
2.New code store into my HOME PC

Now work for Branch

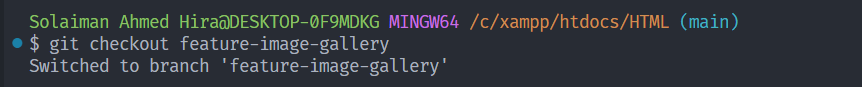
1. **Creating a Branch**:



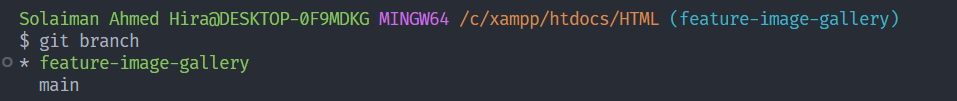
2. **Viewing All Branches**:



3. **Switching Branches**:

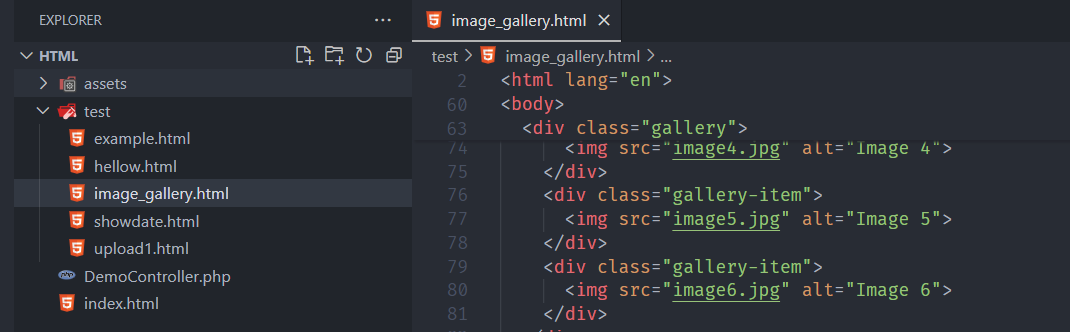


4.View(\* means we are in that branch)

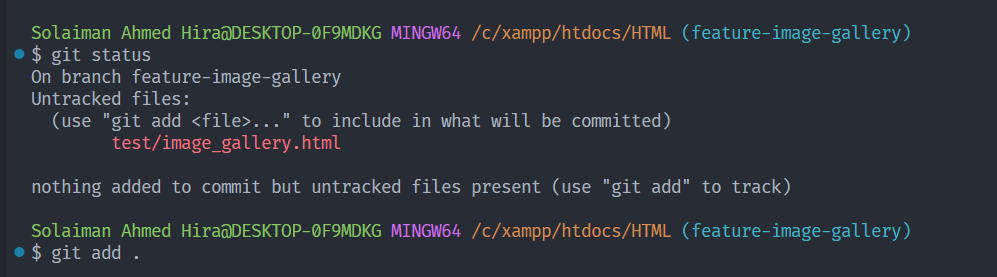


5.Now add new file or work in new branch

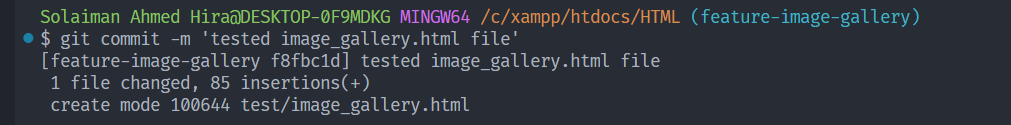
Added image\_gallery.html file



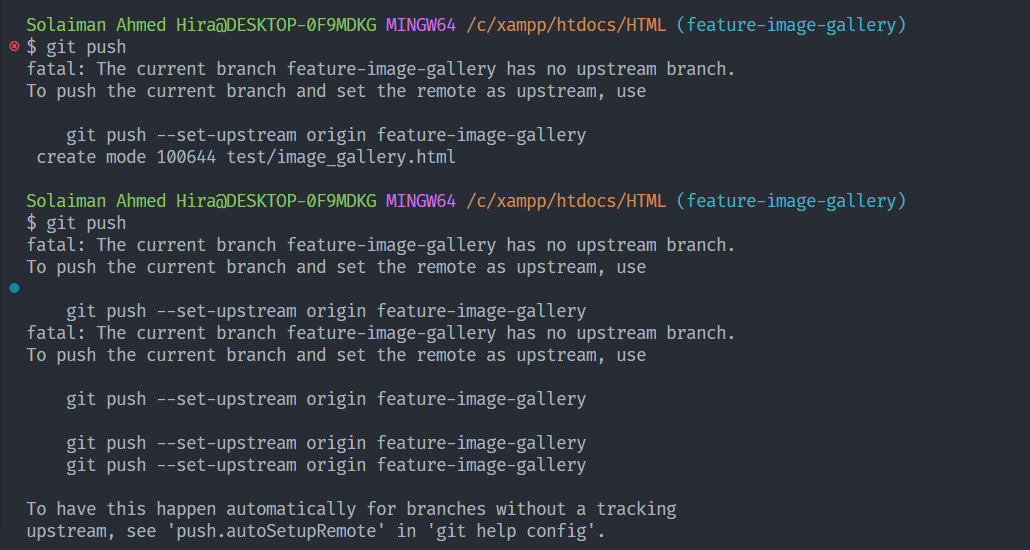
Added image\_gallery.html file to staging area



6.commit this into new branch

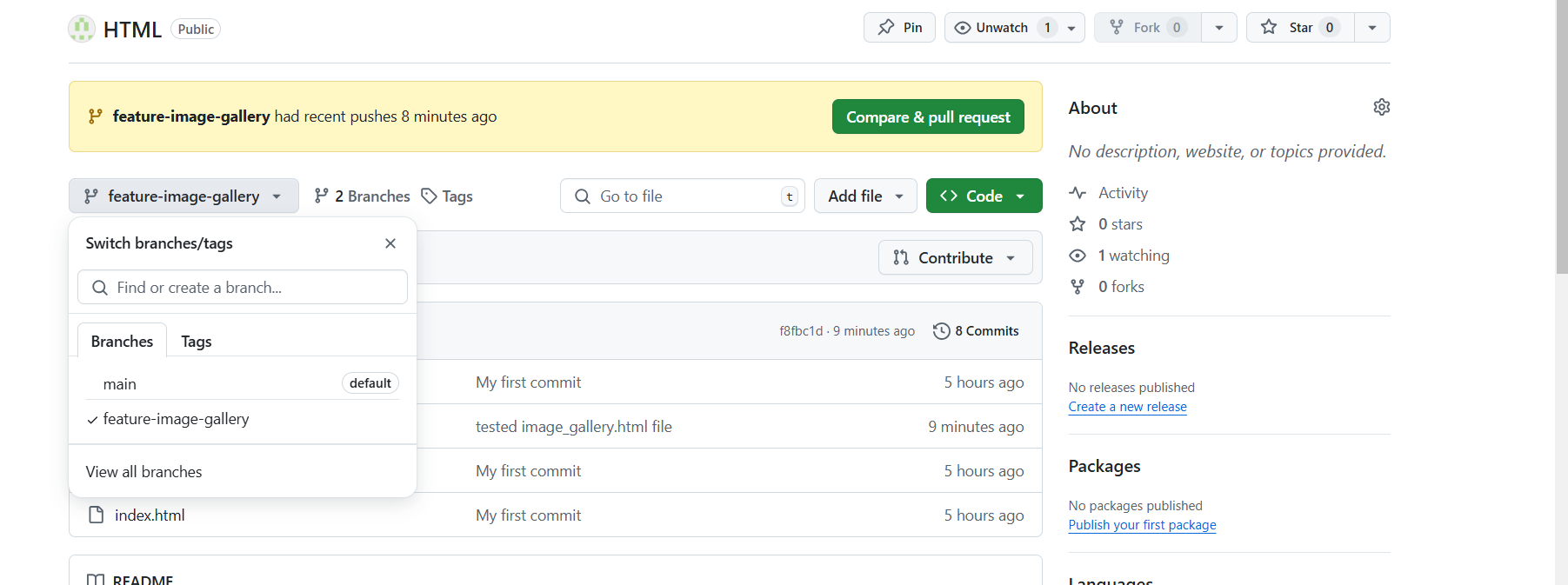


7.Push this new commit to new branch





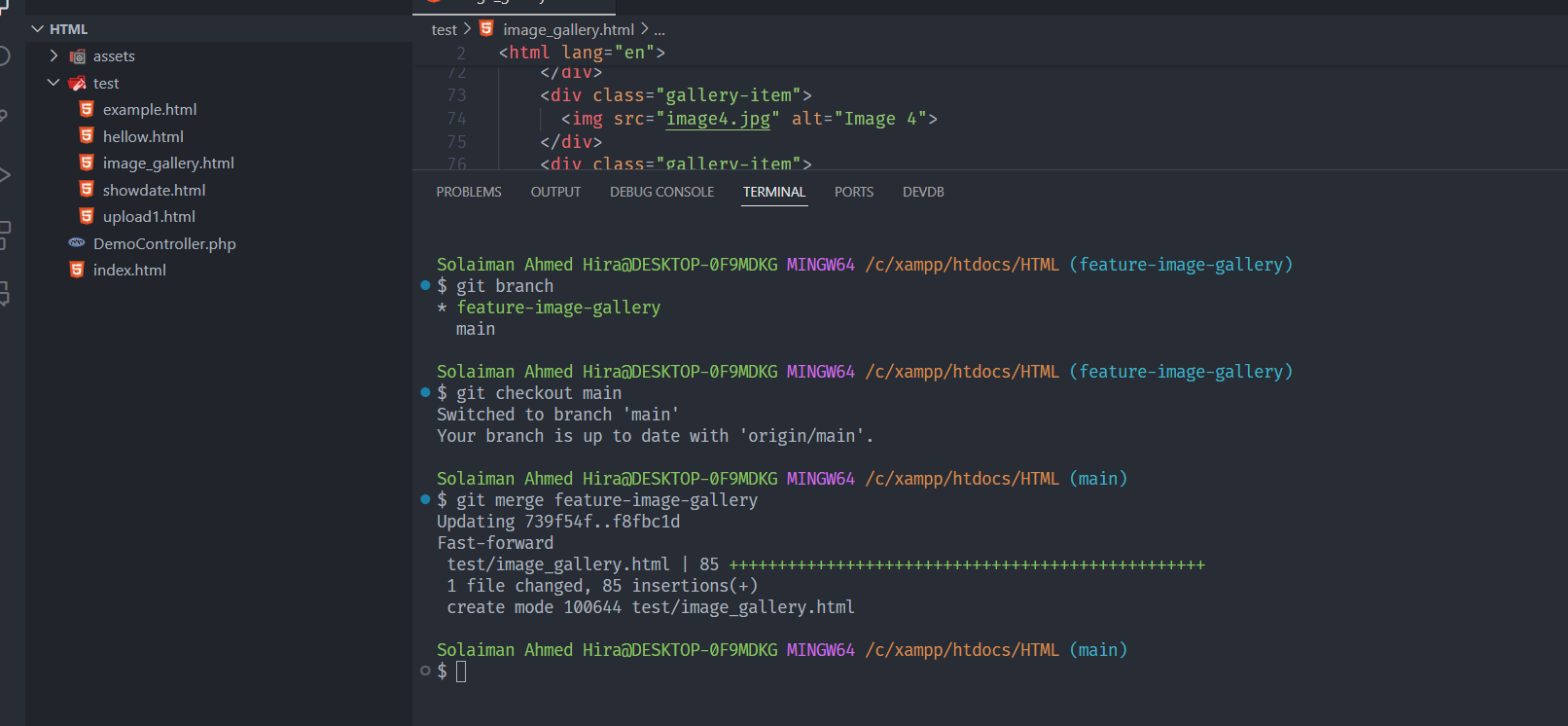
8.New Branch

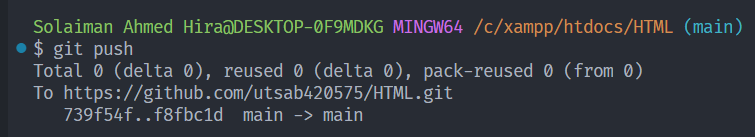


9. **Merge the Branch**: Once the feature is complete and tested, merge it back into main.

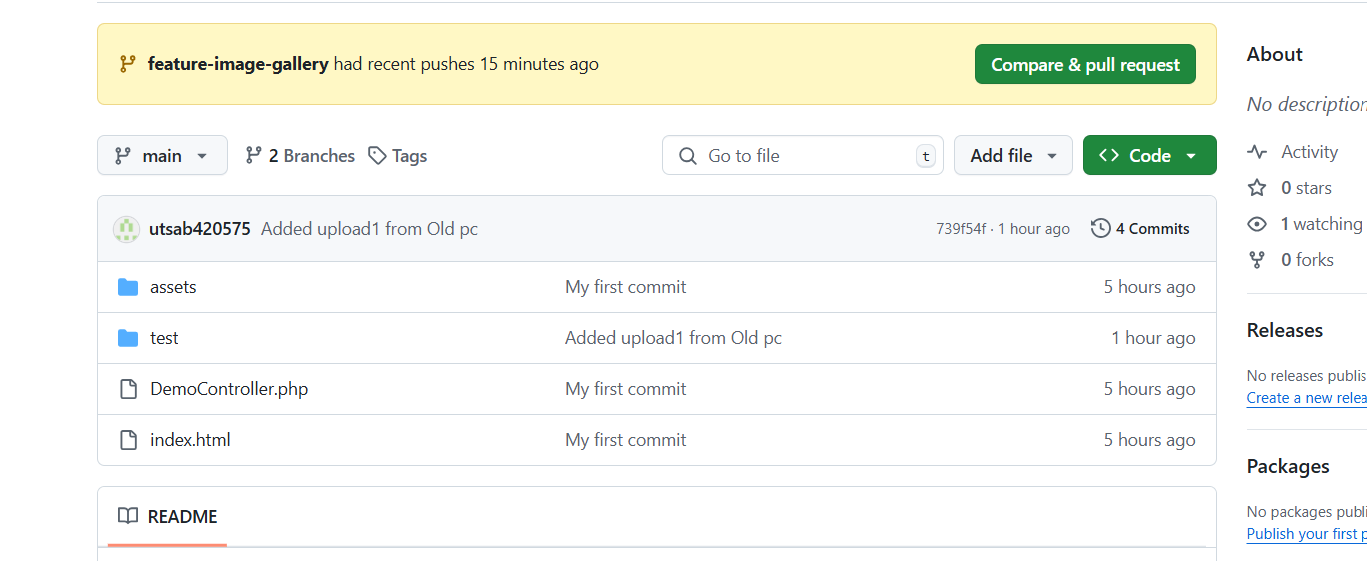
After I think new branch code is ok which is added by another developer. Now we need to merge with main branch.

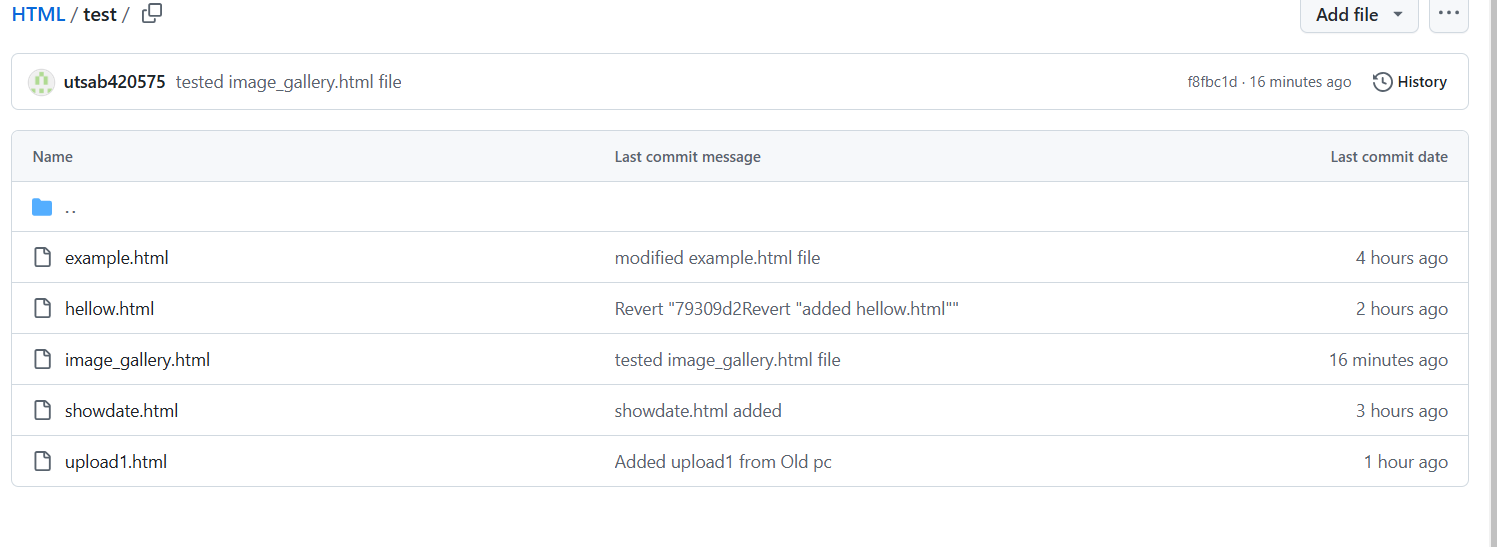
We see that image\_gallery.html is in now main branch.





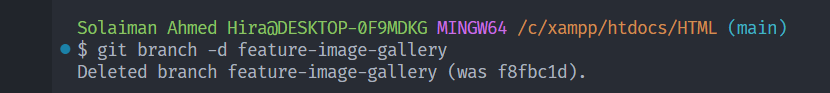
In main branch image\_gallery.html is also uploaded.





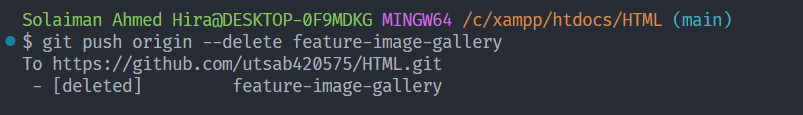
**10.(optional)** Once you’ve merged a branch, you can delete it: **Delete the Feature Branch**: Clean up by deleting the branch.

Delete the Local Branch  
git branch -d feature-image-gallery



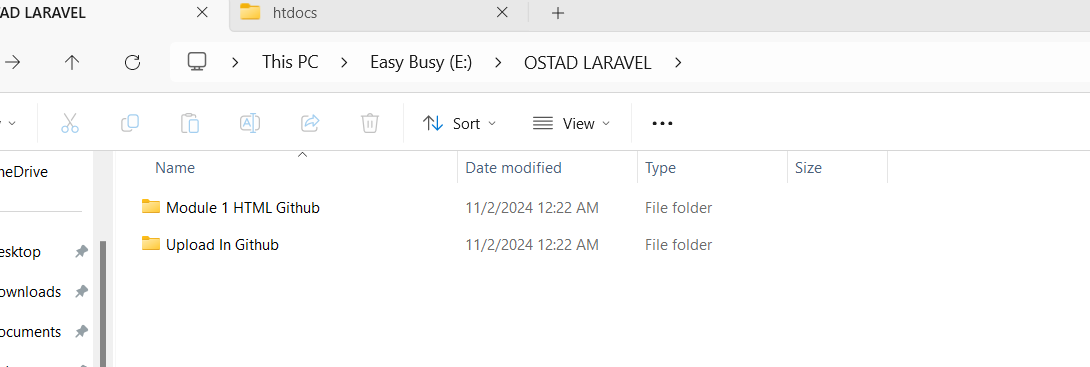
If you want delete branch from github also

git push origin --delete feature-image-gallery

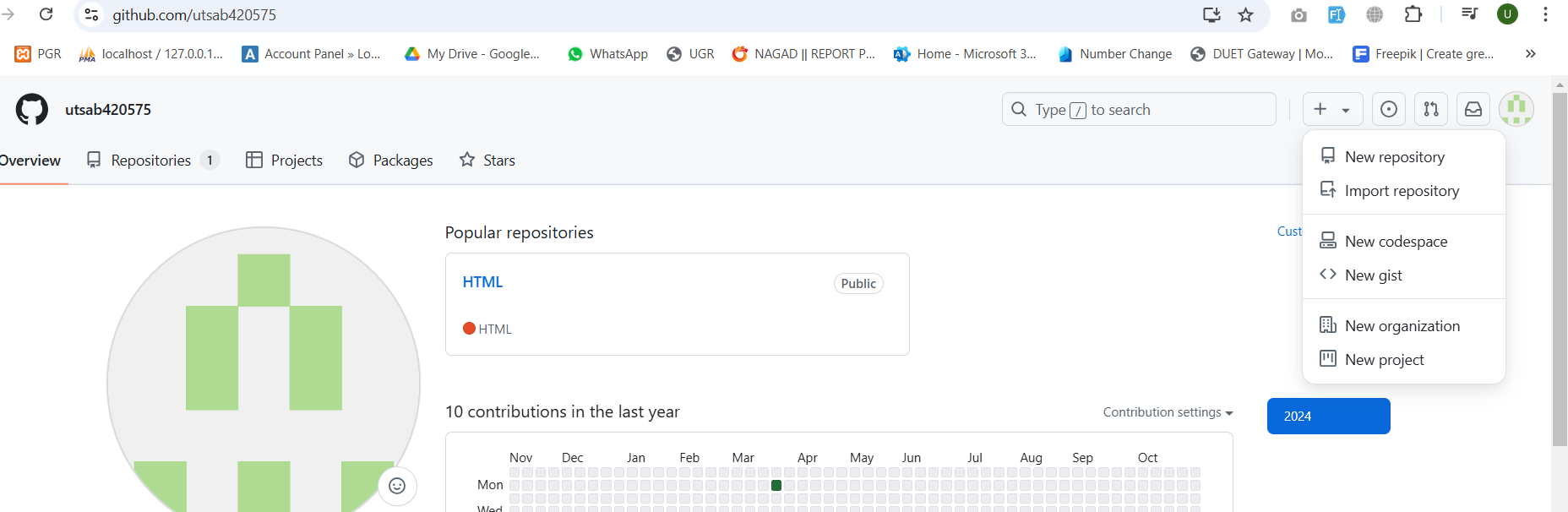


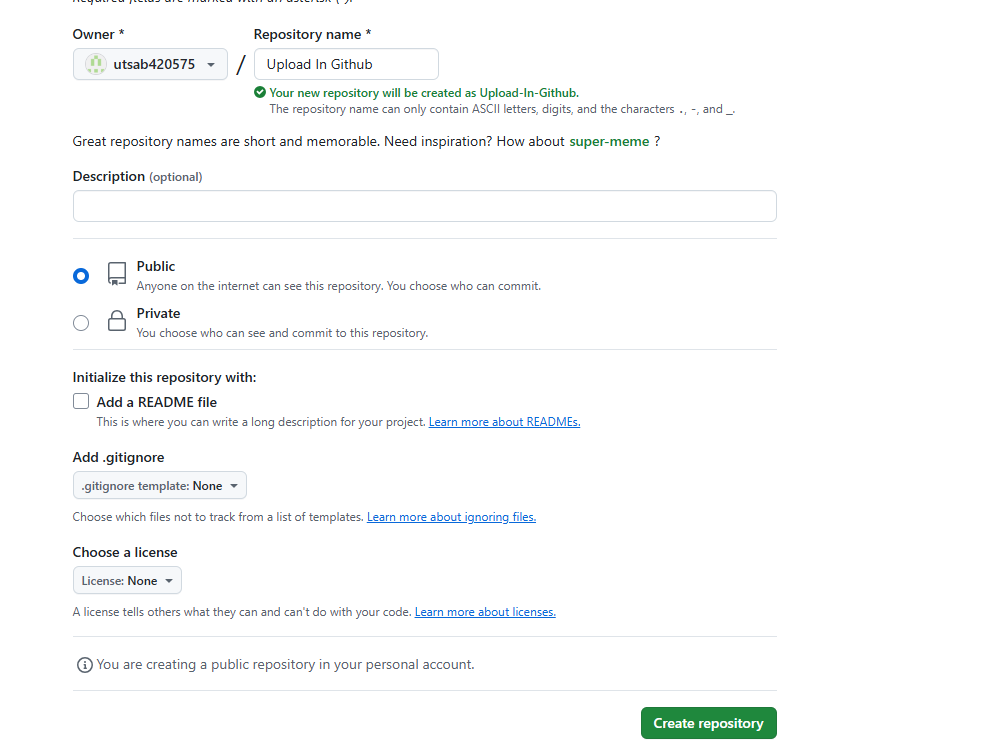
Final Example

1. Create folder Upload In Github   
this folder content I want to upload into github

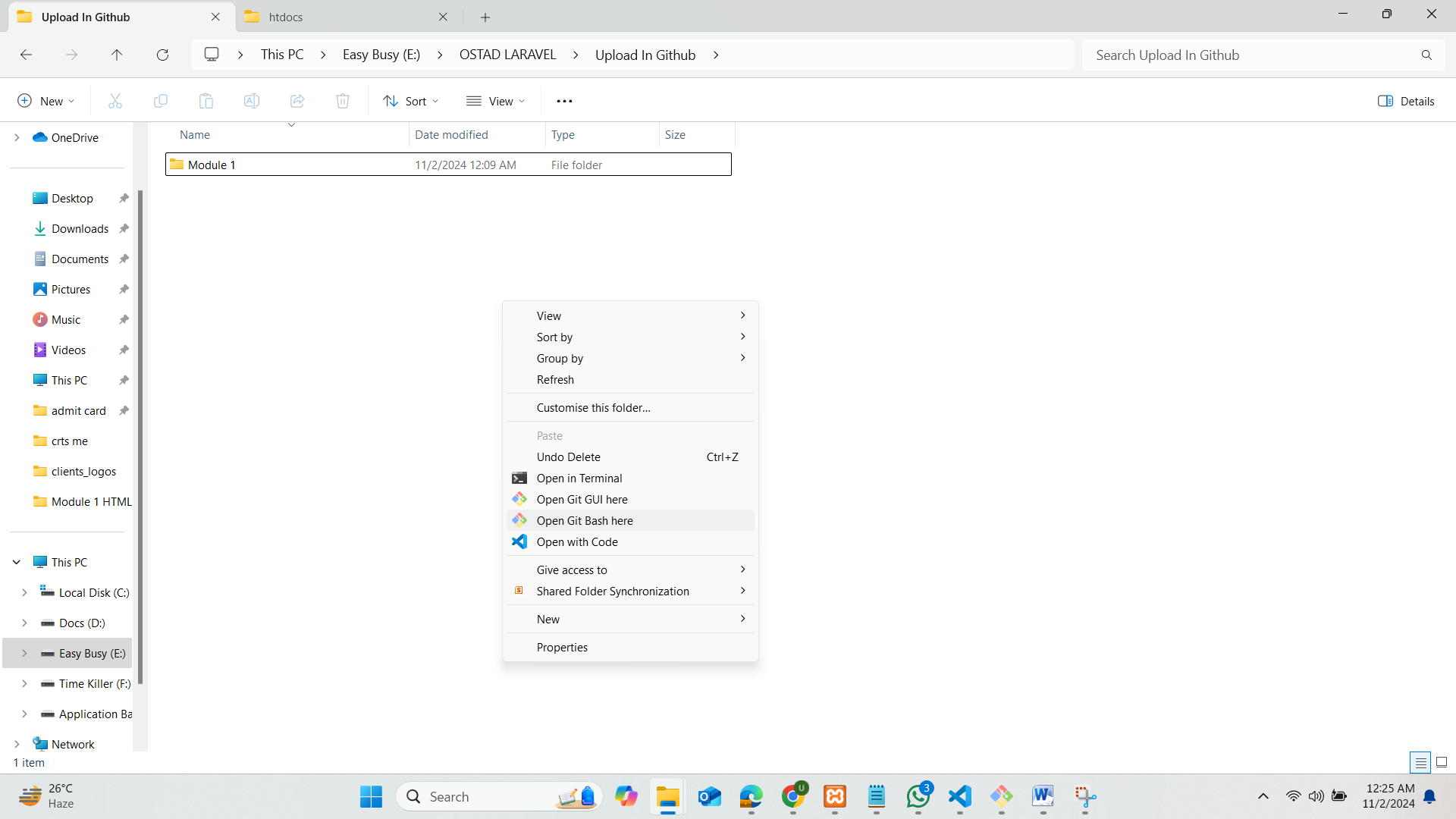


2.Create new repository with same name





3.Go to inside the folder and click right button and open git bash here



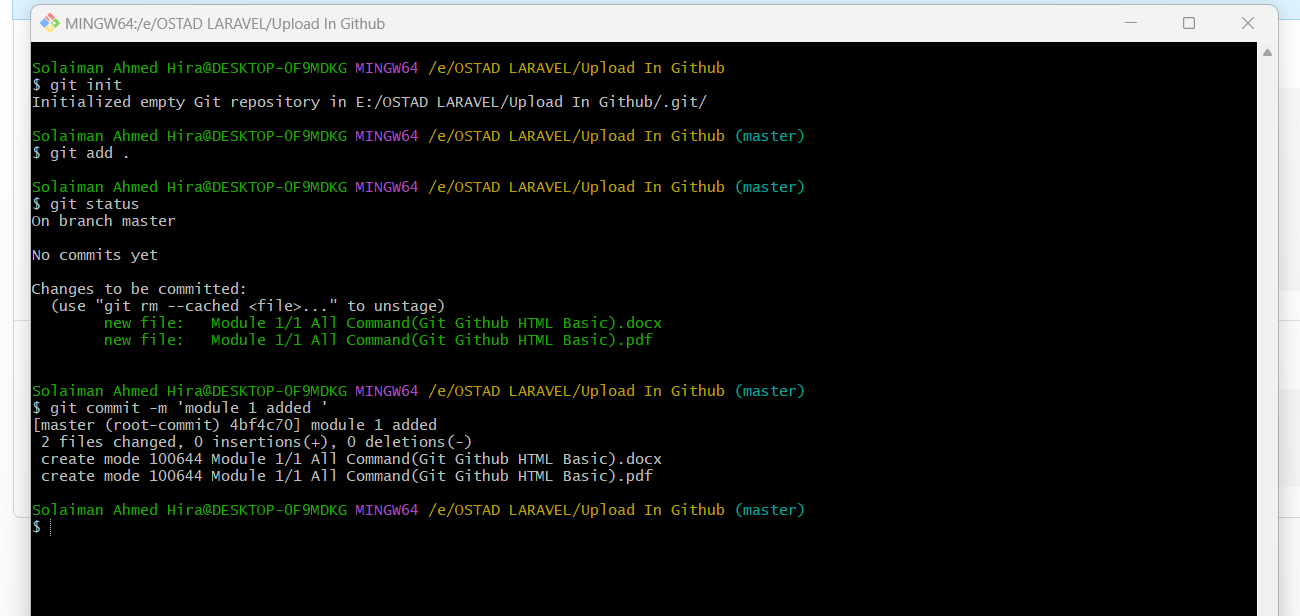
4.Write this code for store into locally

git init

git add .

git status

git commit –m ‘module 1 added’

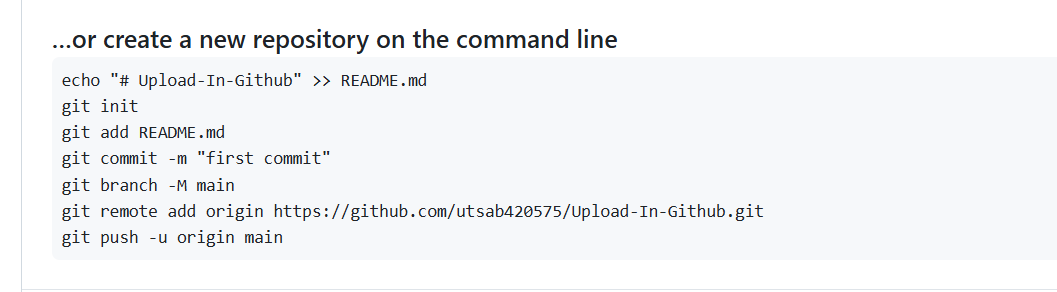


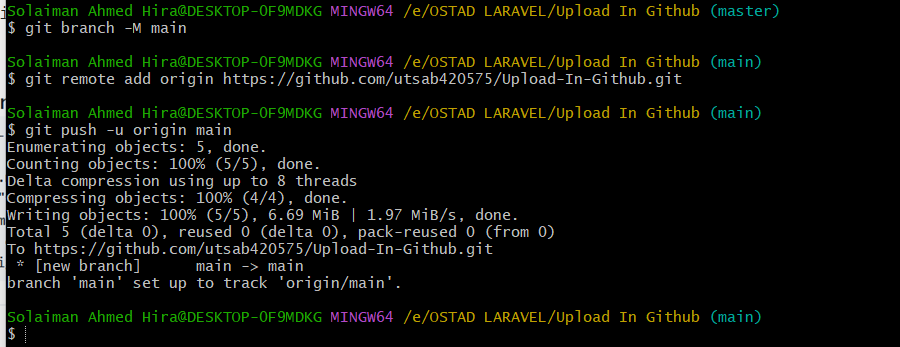
5.Now store into globally(github)

git branch -M main

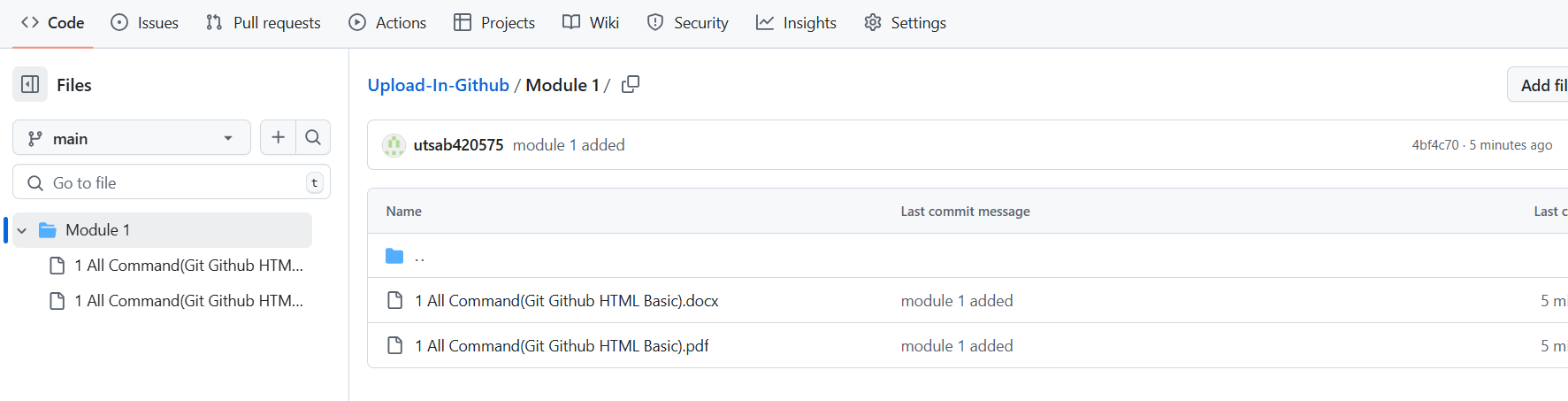
git remote add origin https://github.com/utsab420575/Upload-In-Github.git

git push -u origin main





6.Now data is push into github



//If you modified file and want to upload

