https://omienet.github.io/eportfolio/index.html

Git hub Instructions

**Step 1: Open your ePortfolio folder in VS Code**

* Make sure the folder you want to connect is open in VS Code.
* Open the **terminal** in VS Code (Ctrl + `` on Windows/Linux, Cmd + `` on Mac).

**Step 2: Initialize Git (if not already initialized)**

Run:

git status

* If it says **“not a git repository”**, initialize it:

git init

**Step 3: Add the GitHub remote**

Connect your local folder to your GitHub repository:

git remote add origin https://github.com/omienet/eportfolio.git

* If you get an error that a remote already exists, update it instead:

git remote set-url origin https://github.com/omienet/eportfolio.git

* Verify the remote:

git remote -v

You should see something like:

origin https://github.com/omienet/eportfolio.git (fetch)

origin https://github.com/omienet/eportfolio.git (push)

**Step 4: Pull any existing changes**

To make sure your local folder is in sync with GitHub:

git pull origin main --allow-unrelated-histories

Replace main with your branch name if it’s different. This avoids conflicts if files exist both locally and on GitHub.

**Step 5: Stage and commit your changes**

Whenever you add or edit pages:

git add .

git commit -m "Your commit message here"

Example:

git add security-risk-management.html

git commit -m "Add Security & Risk Management page"

**Step 6: Push to GitHub**

Send your changes online:

git push -u origin main

After this, your local folder and GitHub repository are fully connected. You can now **add, edit, and push any ePortfolio pages**.

**Optional: Preview your changes locally**

* Install the **Live Server** extension in VS Code.
* Right-click any HTML file → **Open with Live Server**.
* Your ePortfolio will open in a browser and update as you save changes.

Test Changes locally before committing

**1. Create a separate branch for testing**

Never make big changes directly on main. Instead, create a branch:

git checkout -b test-changes

* test-changes is just a name—you can call it anything.
* You’re now on a new branch, isolated from main. Any changes here won’t affect your live site.

**2. Make your changes**

* Edit your HTML, CSS, or JS files freely.
* Save your changes locally.
* Preview them in a browser by opening the HTML file directly or using a local server (e.g., VS Code’s **Live Server extension**).

**3. Test locally**

* Open the project in your browser with the **Live Server extension** in VS Code.
* Check that everything looks correct.
* This ensures your website works as expected **before pushing** to GitHub.

**4. Commit your changes in the test branch**

git add .

git commit -m "Tested changes to Security and Risk Management page"

**5. Merge to main only if everything works**

Switch back to main and merge your tested branch:

git checkout main

git merge test-changes

Then push to GitHub:

git push origin main