SIA101: Midterm Exam
1st Semester 2024-2025

Samar State University
SIA101: System Integration and Architecture 1
Units: 3 units

Hosting a Web System on GitHub Pages

Overview

GitHub Pages allows you to host static websites directly from a GitHub repository. These websites can be built using HTML, CSS, and JavaScript, and you can easily link them to custom domains if needed. GitHub Pages can also be integrated with CI/CD (Continuous Integration/Continuous Deployment) pipelines for automated deployments.

Objectives

By the end of this guide, you will:

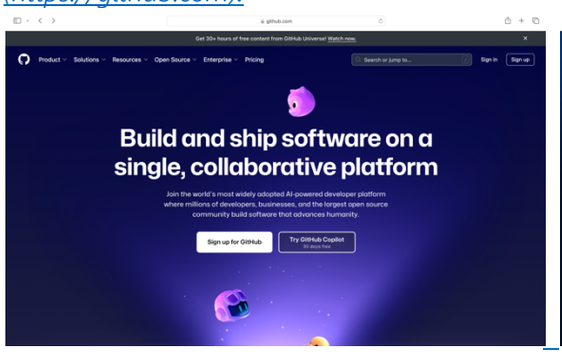
1. Create a GitHub repository for your project.
2. Push your web system (HTML, CSS, JavaScript, etc.) to the repository.
3. Configure the repository to host your site using GitHub Pages.
4. Use Visual Studio Code to write and edit your project files.
5. Access your web system via a public URL (GitHub will provide one by default).

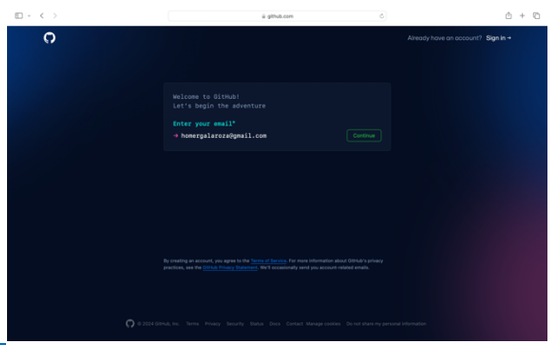
Prerequisites:

Before you begin, ensure you have the following:

1. **Visual Studio Code:** You can download it from here *(https://code.visualstudio.com).*

2. **GitHub Account:** You'll need a GitHub account. If you don't have one, sign up here *(https://github.com).*





3. **Git Installed:** You must have Git installed on your local machine to push your files to GitHub. Download and install it from here *(https://git-scm.com/downloads).*

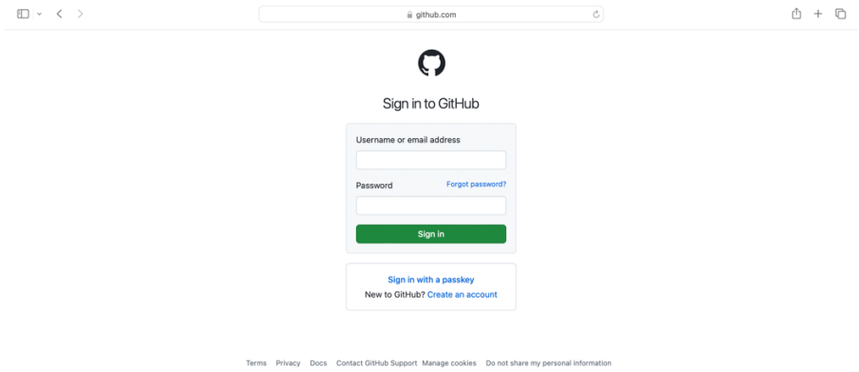
4. **A Web System:** You should have a basic web system (HTML, CSS, JavaScript files) that you want to host. GitHub Pages can only host static websites, meaning no server-side code (like PHP, databases, etc.) will work.

Activity Steps:

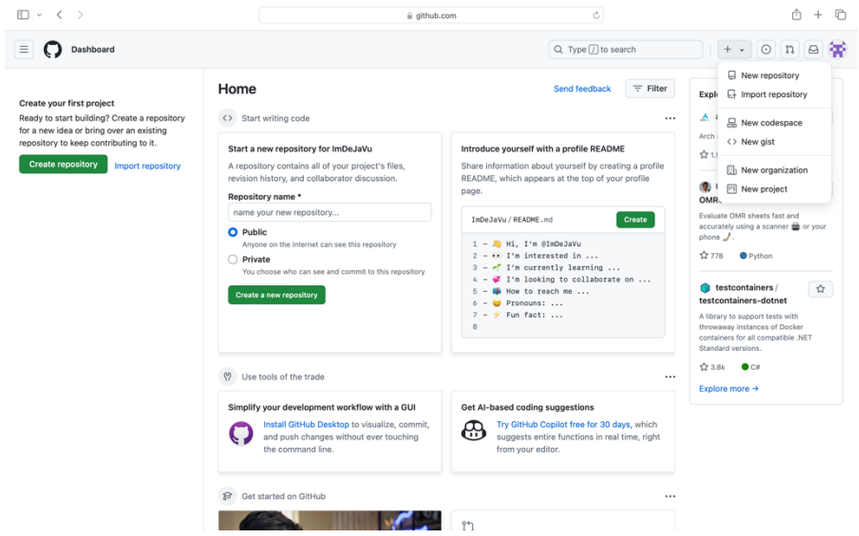
1. **Set Up Your GitHub Repository**

   Before pushing any code, you need a GitHub repository to store your project files.

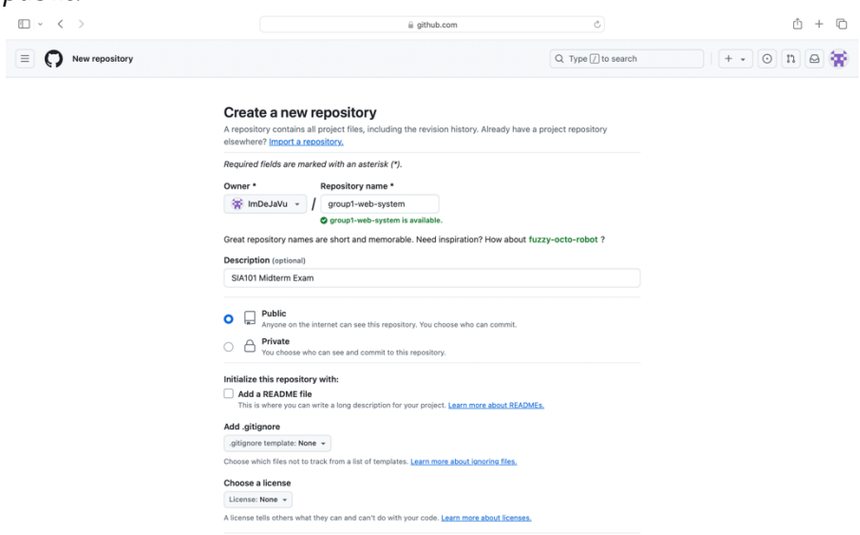
   - Create a New Repository on GitHub:
     - Go to GitHub.



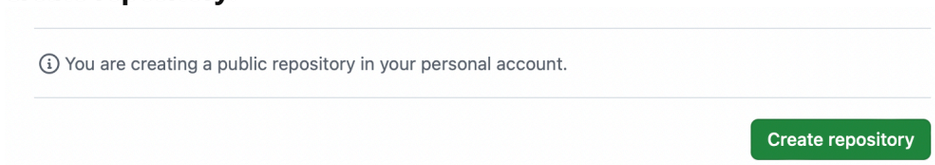
     - Click on the **+ icon** in the top-right corner and select **New repository**.



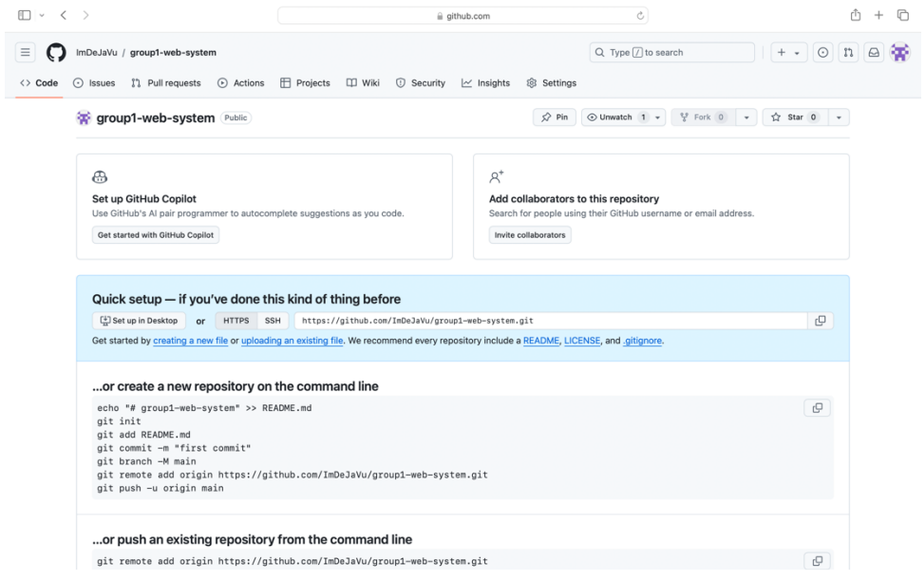
     - Name your repository (e.g., group1-web-system) and add Description (e.g., SIA101 Midterm Exam).
     - Choose whether you want it to be **public** or private. *But this time make it public.*



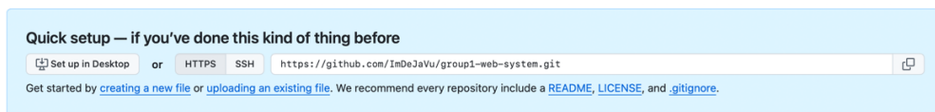
- You can initialize it with a README file, but for this, it's optional. Click **Create repository.**



- Copy Your Repository's URL:
  - After creating the repository, you will be redirected to the repository's page.

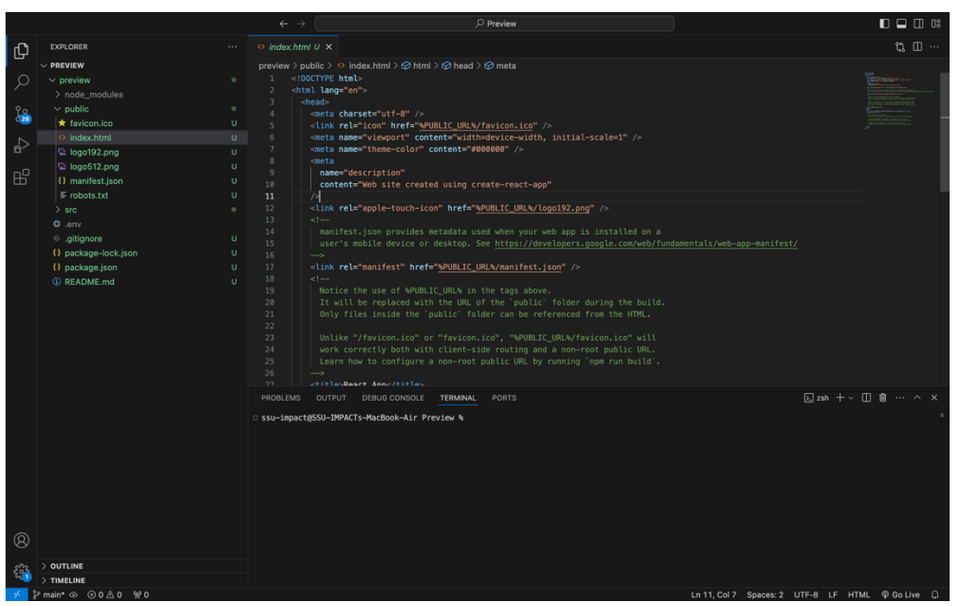


  - Copy the repository URL, which looks like https://github.com/ImDeJaVu/group1-web-system.git

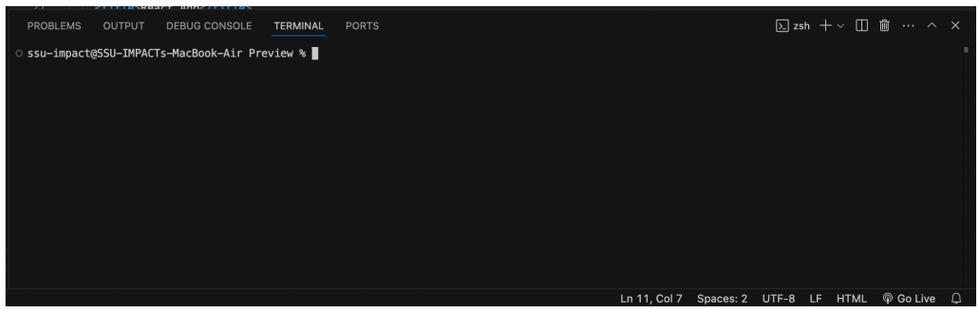


2. Initialize Git in Your Local Project Folder

- Open **VS Code** and open an existing project folder that you want to push to GitHub.

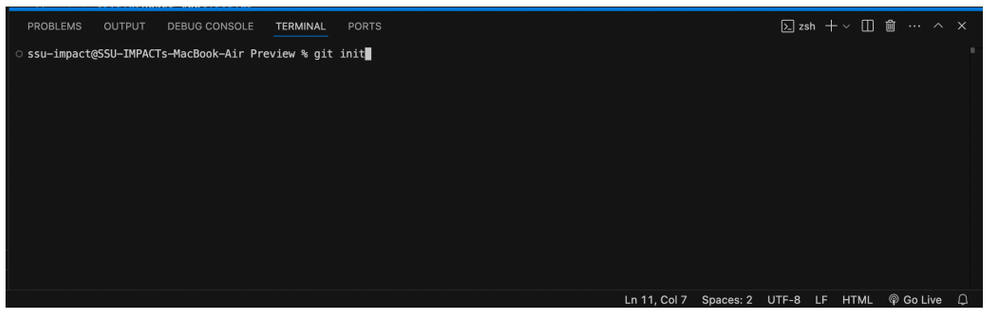


- Open the terminal in VS Code (Ctrl + `) and navigate to your project directory if you haven't already.



- Initialize Git: You must have Git installed on your local machine to push your files to GitHub *(See Prerequisites).*

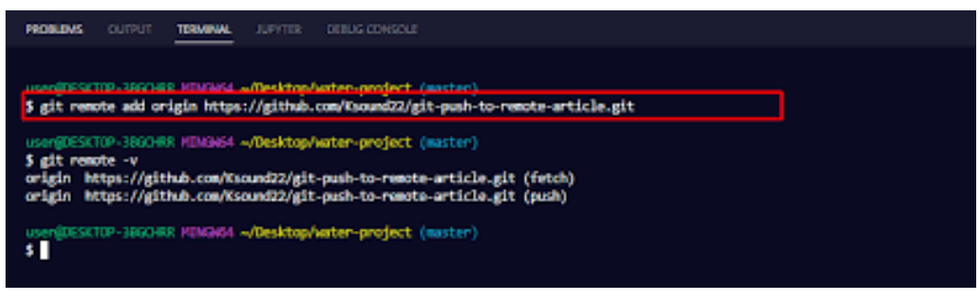
  git init



  This will create a new .git folder that allows you to track your project using Git.

- Add Your Remote Repository: Link your local project to the remote repository you created on GitHub.

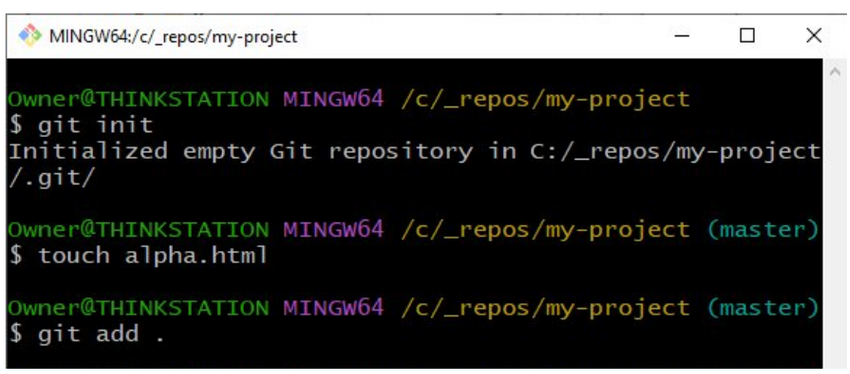
  git remote add origin https://github.com/ImDeJaVu/group1-web-system.git



3. Commit Your Changes

- Track Your Files: Add all your project files to Git for version control:

  git add .



  Note: Don't forget the dot ('.') after the add.

- Commit Your Changes: Create an initial commit to save your changes:

  git commit -m "Initial commit"

```
Owner@THINKSTATION MINGW64 /c/_repos/my-project (master)
$ git commit -m "1st commit"
[master (root-commit) f131bee] 1st commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 alpha.html
```

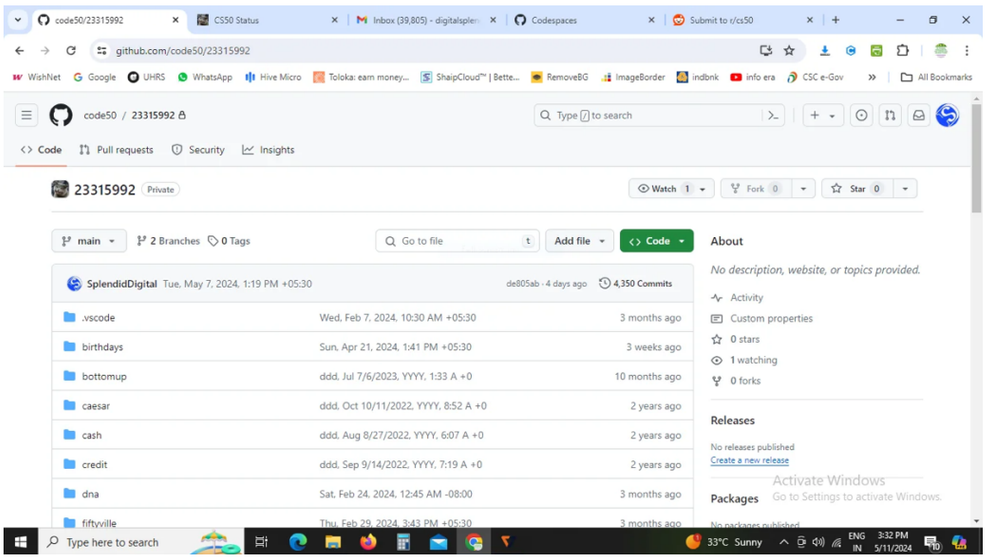
4. **Push Your Code to GitHub**

- Push Your Code: To send your local project to GitHub, use the following command:

  git push -u origin master

  *If you're working with a **main** branch instead of **master**, replace **master** with **main**.*

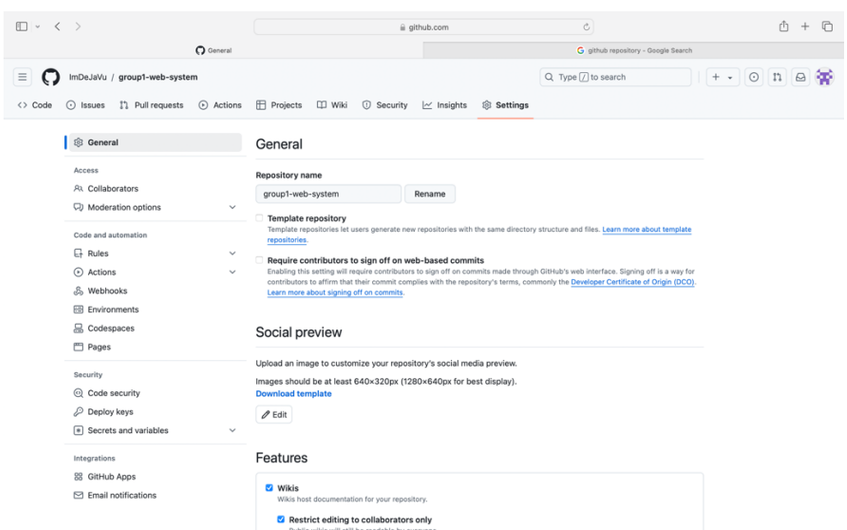
- Verify on GitHub: After pushing, go to your GitHub repository page, and you should see your project files.



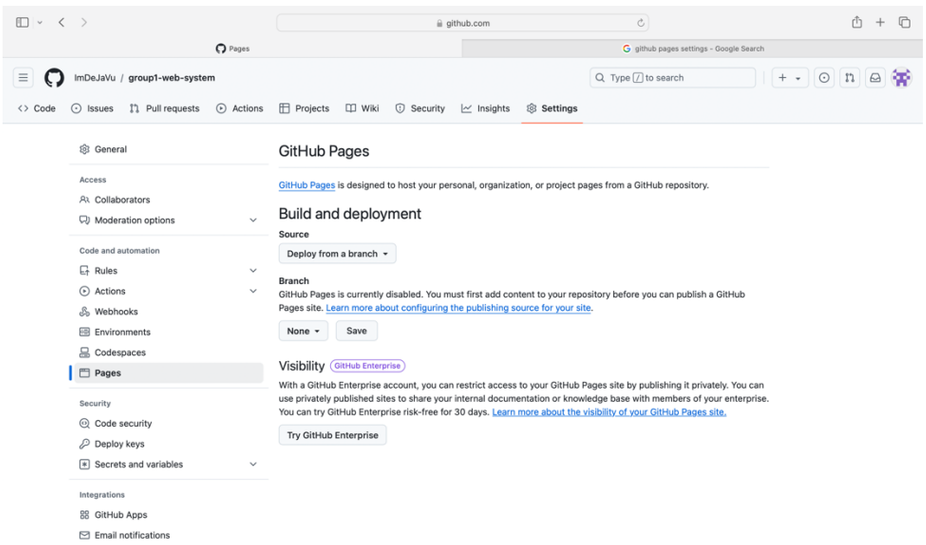
5. Set Up GitHub Pages

If your project is a static website (HTML, CSS, JavaScript), you can use GitHub Pages to host it.

- Go to Your Repository Settings:
  - On **GitHub**, navigate to your repository.
  - Click on the **Settings** tab.
  - Scroll down to the **Pages** section.



- Configure GitHub Pages:
    - Under **Source**, choose the branch you want to use for GitHub Pages. For this guide, select **main** (or **master** if you used that).
    - You can choose the folder from which to serve the files. If you want to serve directly from the root folder, select / (root).
    - Click **Save**.



- Access Your Website: GitHub will provide a URL where your static website is hosted. It will look like this:

    https://username.github.io/repository-name/

    Congratulations! You've successfully hosted your web system on GitHub and set up GitHub Pages to make your website publicly accessible.

By following these steps, you've learned how to use Git for version control and GitHub Pages for free hosting of static websites. Now, whenever you make updates or changes to your project in Visual Studio Code, you can commit and push those changes to GitHub, and they will automatically reflect on your live website. This process not only makes your project accessible to others but also helps you manage and track your code effectively. Keep experimenting with new features and designs, and enjoy the benefits of using GitHub to streamline your development workflow!

(Optional Steps for updating the website)

Make Changes and Push Updates

- Edit Files:
    - Use Visual Studio Code to make changes to your project (e.g., editing HTML, CSS, JavaScript files).

- Commit Changes: Every time you make changes, remember to commit and push:

    bash
    Copy code
    git add .
    git commit -m "Updated [feature/bugfix/etc.]"
    git push

- View Updates: Your changes will reflect immediately on your GitHub Pages URL (if your website is hosted there). It may take a minute or two to update.