# Lab Report: Building a Personal Website with GitHub Pages

I. Lab Title

Building and Publishing a Static Personal Website Using GitHub Pages

II. Purpose and Objectives

Understand the concept of static websites and how they differ from dynamic websites.

Master the use of GitHub Pages as a PaaS (Platform as a Service) for hosting static content.

Learn to create, manage, and publish a GitHub repository for web hosting.

Use HTML, CSS, or templates to customize and design a personal website.

Develop professional digital literacy skills relevant to modern software development.

III. Lab Environment

Platform: GitHub (https://github.com)

Service: GitHub Pages

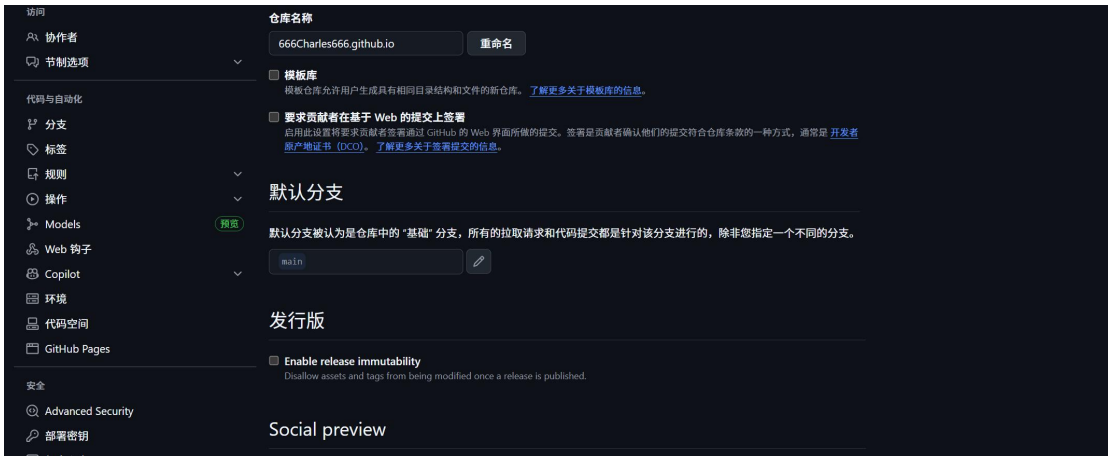
Technology: HTML, CSS, Git Version Control

Equipment: Computer with internet access

IV. Lab Content and Procedures

Step 1: Create a GitHub Repository

As required by the lab, a special repository named after the username was created. The repository is named [YourGitHubUsername].github.io. This repository will serve directly as the root directory of the website.

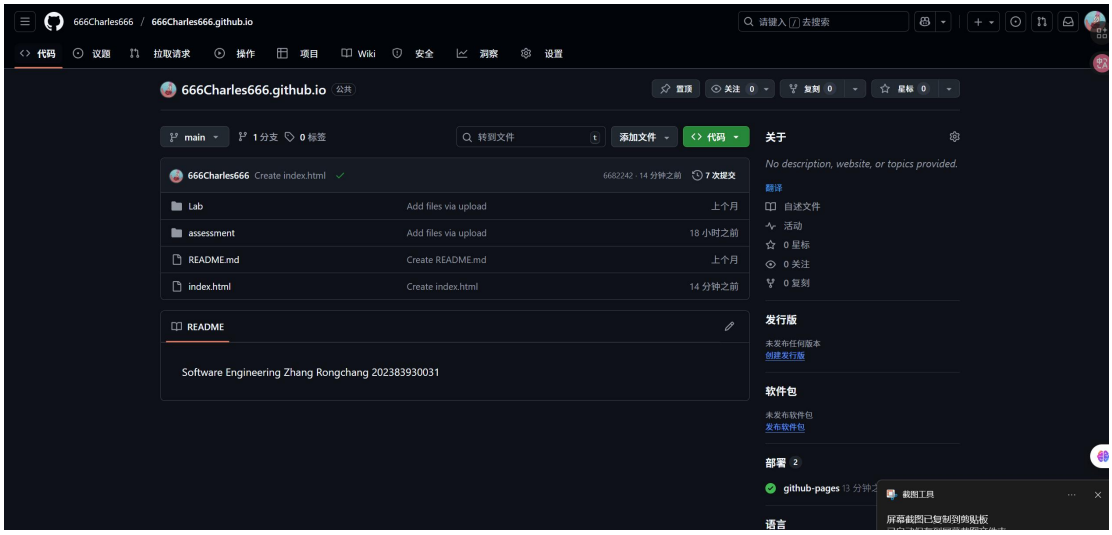



Step 2: Create Website Files

Create the core website files locally or via the GitHub web interface. This must include at least an index.htmlfile as the homepage of the website.

Step 3: Write Website Content

Use HTML and CSS to create a personal introduction homepage and apply personalized styling. Content may include self-introduction, skills, project experience, or blog links.





Step 4: Commit and Push Code

Use Git commands or the GitHub web interface to commit the website files to the main branch (default is mainor master).

Step 5: Configure and Publish the Website

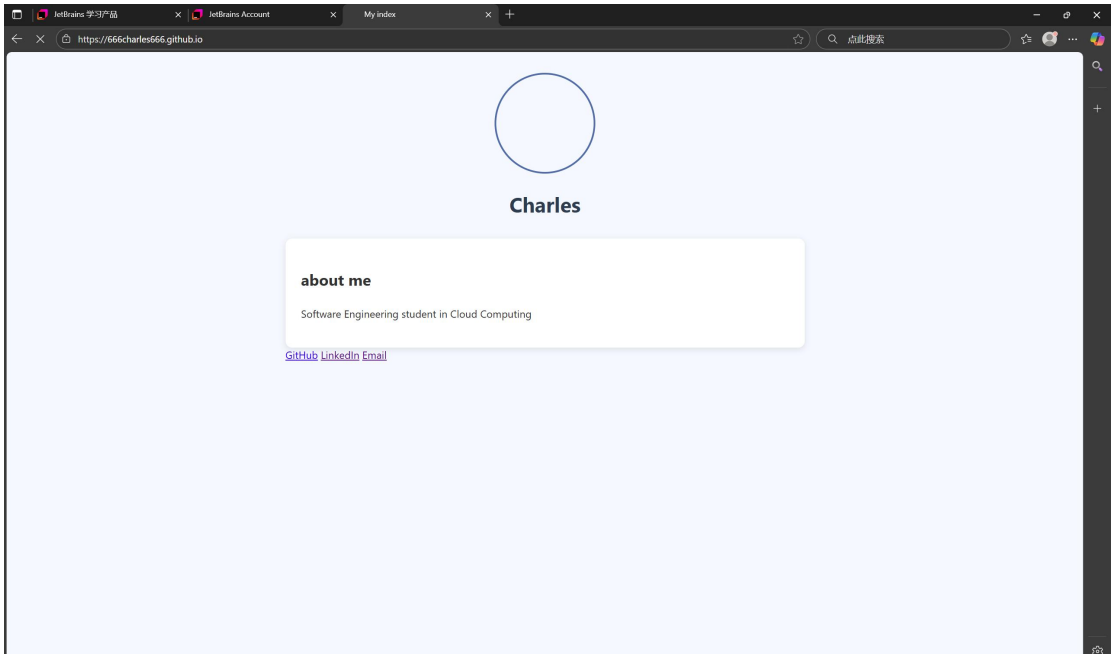
In the repository's Settings, find the "Pages" option, set the source branch to the deployment branch (usually the mainbranch), and confirm the website is successfully published.

Screenshot 3: Successful GitHub Pages configuration page

(Paste your screenshot of the Pages section in your repository Settings here, showing the green message "Your site is published at...")

Step 6: Access and Verify

Enter https://666charles666.github.io in a web browser address bar to access the published website and verify that it is accessible and displays correctly.




V. Results

GitHub Repository URL:

https://github.com/666Charles666/666Charles666.github.io.git

Personal Website URL:

https://666charles666.github.io

Description of Outcomes:

Successfully created and deployed a static personal website based on GitHub Pages. The website includes a self-introduction homepage and has been personalized with custom design and styling. Through this lab, the complete process of using Git for version control and utilizing the GitHub Pages service for automated deployment was mastered.

VI. Summary and Reflections

Through this experiment of "Building a Personal Website with GitHub Pages", I successfully completed the entire process of setting up, deploying and launching a static website from scratch to completion, and gained a lot.

First of all, I have a deep understanding of the basic concepts of static websites and the working principle of GitHub Pages as a static website hosting service (PaaS). Its greatest advantage lies in the seamless integration of version control and automated deployment. Developers only need to focus on content creation and code writing, without having to worry about the cumbersome server configuration, maintenance and expansion issues. This greatly lowers the technical threshold for web publishing.

Secondly, this experiment has enabled me to gain a more solid grasp of the practical application of Git and GitHub. From creating a repository, committing code to finally pushing it to the remote repository, I have fully practiced the basic version control process. This not only ensures the version management of project code but also familiarizes me with the indispensable collaboration tools in modern software development.

During the practice process, I also encountered and resolved some challenges. For instance, when customizing CSS styles, I faced the issue of page layout misalignment at different browser window sizes. By consulting resources, I learned the basic concepts of responsive web design and attempted to make initial adjustments using percentage widths and media queries. Eventually, I made the page layout more flexible. Although this process took time, it greatly enhanced my ability to identify, analyze, and solve problems with the help of resources.

Finally, the outcome of this experiment is not only a technical exercise but also a valuable personal asset. This website can serve as my online resume and portfolio. In the future, I will continue to maintain and update it, adding more projects and content, making it an important showcase for my personal brand and career development.

In conclusion, this experiment was a highly successful learning experience that combined theory with practice. It gave me a more intuitive and in-depth understanding of front-end development, version control, and automated deployment, effectively enhancing my practical skills and digital literacy, and laying a solid foundation for my future studies and development.