**Report for Publishing a Website Using GitHub Pages and configuring a Website in a LAN Network Using XAMPP and access XAMPP Server from Another Computer on the Same Network**

Submitted by

**NANDINI SAIN**

**INTERN**

**3RDITECH**

**Acknowledgement**

I would like to thank Ms Vrinda Kapoor, Founder of 3rditech for giving me the opportunity to do this internship within the organization.

I further thank Mr. Aakash Sir and Ritwik Sir for guiding and teaching me throughout the entire process. These skills would help me to expand my resume and advance my career.

**Information about the Internship position**

I joined 3rditech in the capacity of an Intern, at Research and Innovation Park, IIT Delhi, New Delhi. This internship was done in lieu of my status as a final year undergraduate student at Amity University, Noida.

I worked under the Embedded Systems team which was led by Aakash sir. My main tasks were to design algorithms for the code, do connections, collect data and do research relevant information about the project that I was assigned to.

**Table of Contents**

Part-1- Publishing a Website Using GitHub Pages

1. Introduction
2. Prerequisites
3. Step-by-Step Guide
   1. Creating a GitHub Repository
   2. Uploading Website Files
   3. Enabling GitHub Pages
4. Verification
5. Troubleshooting
6. Conclusion
7. References

Part-2

1. Introduction
2. Prerequisites
3. Step-by-Step Guide
4. Installing and Starting XAMPP
5. Setting Up Your Website
6. Configuring Apache to Listen on All Network Interfaces
7. Allowing Apache Through Windows Firewall
8. Accessing the Website from Another Device on the LAN
9. Verification
10. Troubleshooting
11. Conclusion
12. References

**Part-1-Publishing a Website Using GitHub Pages**

**1. Introduction**

This document covers the steps and aspects undertaken to connect a website and make it publicly available, in case of unavailability of domain servers.

**2. Prerequisites**

List the tools and accounts needed:

* A GitHub account.
* Basic understanding of Git and GitHub
* A sample website project

**3. Step-by-Step Guide**

**A. Creating a GitHub Repository**

1. **Log into GitHub**: Open [GitHub](https://github.com/) and log into your account.
2. **Create a New Repository**:
   * Click on the **+** icon at the top right and select **New repository**.
   * Name your repository (e.g., **my-website**).
   * Choose the repository's visibility (public or private).
   * Initialize the repository with a README file (optional).
   * Click **Create repository**.

**B. Uploading Website Files**

1. **Clone the Repository** (if using Git):

sh

Copy code

git clone <https://github.com/your-username/my-website.git>

1. **Add Your Website Files**:
   * Copy your website files (HTML, CSS, JS, images, etc.) into the cloned repository folder.
   * You can directly also upload / add an existing file to the github repository folder.
2. **Commit and Push Changes**:

* sh

Copy code

cd my-website git add . git commit -m "Initial commit" git push origin main

* In case you are unfamiliar with Git, you can commit changes directly on GitHub.

**C. Enabling GitHub Pages**

1. **Go to Repository Settings**:
   * Navigate to the repository on GitHub.
   * Click on **Settings**.
2. **Enable GitHub Pages**:
   * Scroll down to the **GitHub Pages** section.
   * Under **Source**, select the branch (e.g., **main**) and folder (e.g., **/root**).
   * Click **Save**.
   * GitHub will provide a URL where your website is now live.

**4. Verification**

* **Access Your Website**:
  + Open the provided GitHub Pages URL in a browser to ensure the site is live.
  + If using a custom domain, ensure it correctly redirects to your GitHub Pages site.

**5. Troubleshooting**

* **Common Issues**:
  + Changes not appearing: Ensure you have committed and pushed all changes.
  + Custom domain not working: Double-check DNS settings.
  + 404 errors: Verify file paths and GitHub Pages settings.

**6. Conclusion**

Publishing a website with GitHub Pages is easy and effective. You create a repository on GitHub, upload your website files, enable GitHub Pages in the settings, and your site is live. You can also set up a custom domain if you want.

**Benefits of GitHub Pages:**

* Free Hosting: No cost to host your static website.
* Easy Updates: Changes you make are automatically shown on your site.
* Custom Domain: You can use your own web address.
* Reliable and Fast: Uses GitHub's infrastructure for good performance.
* Using GitHub Pages is a great way to host your website for free, with simple updates and reliable performance.

**7. References**

* [GitHub Pages Documentation](https://docs.github.com/en/pages)
* [GitHub Repository Creation](https://docs.github.com/en/get-started/quickstart/create-a-repo)

**Part-2- Configuring a Website in a LAN Network Using XAMPP and access XAMPP Server from Another Computer on the Same Network**

**Introduction**

This documentation provides a comprehensive guide on how to configure a website on a LAN network using XAMPP, enabling access from other devices within the same network.

**2. Prerequisites**

* A computer with XAMPP installed.
* Website files ready to be hosted.
* Basic knowledge of network settings.
* Administrator access to the host computer.

**3. Step-by-Step Guide**

**A. Installing and Starting XAMPP**

1. **Download XAMPP** from the official website and install it on your host computer.
2. **Open the XAMPP Control Panel** and start the Apache service.

**B. Setting Up Your Website**

1. Place your website files (e.g., **index.html**, **index.php**) in the **htdocs** directory, typically located at **C:\xampp\htdocs**.

**C. Configuring Apache to Listen on All Network Interfaces**

1. **Edit the httpd.conf file**:
   * Navigate to **C:\xampp\apache\conf\httpd.conf**.
   * Change **Listen 80** to **Listen 0.0.0.0:80**.
   * Ensure **ServerName 0.0.0.0:80** is set correctly.

**D. Allowing Apache Through Windows Firewall**

1. **Open Windows Defender Firewall settings**:
   * Control Panel > System and Security > Windows Defender Firewall > Allow an app or feature through Windows Defender Firewall.
   * Click "Change settings".
   * Find and enable "Apache HTTP Server" for both "Private" and "Public" networks.
   * If not listed, add **httpd.exe** from **C:\xampp\apache\bin\**.

**E. Accessing the Website from Another Device on the LAN**

1. On another device connected to the same network, open a web browser.
2. Enter the host computer's IP address in the address bar, followed by the path to your site content:

plaintext

Copy code

http://<host\_computer\_ip>/index.html

**4. Verification**

* Open a web browser on another device within the same network.
* Navigate to the host computer's IP address (e.g., **http://192.168.0.121/index.html**) to verify the website is accessible.

**5. Troubleshooting**

* **Firewall Settings**: Ensure the firewall on the host computer allows incoming connections on the configured port.
* **Network Connectivity**: Verify both devices are on the same network and can ping each other.
* **Port Conflicts**: If using a port other than 80, specify the port in the URL (e.g., [**http://192.168.0.121:8080/index.html**](http://192.168.0.121:8080/index.html)).

**6. Conclusion**

By following the steps outlined in this guide, you can successfully configure a website on a local computer using XAMPP and make it accessible to other devices within the same local network.

**7. References**

* XAMPP Official Website
* [Apache HTTP Server Documentation](https://httpd.apache.org/docs/)
* [Windows Firewall Documentation](https://support.microsoft.com/en-us/windows/windows-defender-firewall-helps-protect-your-pc-97bb81aa-29b3-8ac3-86fc-3e5632f19b63)