Server

<https://secure.sakura.ad.jp/rs/cp/domain/list>

Released

<https://glanzconsulting.sakura.ne.jp/main/>

Certainly! You can achieve this using GitHub Pages, which is a feature provided by GitHub that allows you to host static websites directly from your GitHub repository. Here's a step-by-step guide to help you think like a developer and get your HTML/CSS files hosted on GitHub Pages:

1. \*\*Create a GitHub Repository:\*\*

- If you haven't already, sign up for a GitHub account.

- Create a new repository. Give it a meaningful name, like "client-demo-site".

2. \*\*Prepare your local workspace:\*\*

- Organize the HTML/CSS files in a folder on your local machine. Ensure your main page is named `index.html` as this will be the default page GitHub Pages will look for when serving your website.

3. \*\*Push your files to the GitHub Repository:\*\*

- Initialize a Git repository in your local folder:

```bash

git init

```

- Add your files to the Git repository:

```bash

git add .

```

- Commit your changes:

```bash

git commit -m "Initial commit"

```

- Link your local repository to your GitHub repository:

```bash

git remote add origin [your-github-repo-url]

```

- Push your files to GitHub:

```bash

git push -u origin main

```

4. \*\*Setup GitHub Pages:\*\*

- Go to your GitHub repository on the web.

- Click on the `Settings` tab.

- Scroll down to the `GitHub Pages` section.

- Under the "Source" dropdown, select the `master` branch (or main branch if you named it differently).

- You can optionally choose a theme, but it's not mandatory.

- Your site will now be published at a URL similar to: `https://[your-github-username].github.io/[repository-name]/`

5. \*\*Updating the Site:\*\*

- When you want to make changes to your site, simply update your HTML/CSS files on your local machine.

- Once changes are made, repeat the add, commit, and push steps:

```bash

git add .

git commit -m "Description of changes made"

git push

```

- The changes will be reflected in real-time on the GitHub Pages URL.

\*\*Reasoning Behind This Approach:\*\*

- GitHub Pages is designed for hosting static content, which includes HTML, CSS, and JavaScript files. It provides a quick and free solution for showcasing projects, portfolios, or demo sites.

- When you make changes to your repository and push them to GitHub, the GitHub Pages site automatically updates, allowing for real-time feedback.

\*\*Example:\*\* Suppose you made a change to your `index.html` file to update a header text. After pushing this change to GitHub, when you share the GitHub Pages link with your client, they will immediately see the updated header text.

Remember, while GitHub Pages is excellent for static content, it won't handle server-side scripts like PHP or databases. For dynamic content or back-end operations, you'd need to consider a different hosting solution.