

# Report: Running an Assignment Viewer Website Using GitHub Pages

## Introduction

Hosting an Assignment viewer website the usage of GitHub Pages is a trustworthy and efficient manner to proportion documents with a extensive audience. GitHub Pages gives a free and reliable web hosting service for static web sites without delay from a GitHub repository. This record outlines the stairs to installation, set up, and manage a Assignment viewer website using GitHub Pages.

## Setting Up GitHub Pages

1. **Login to GitHub Account:**
  - o First of all, I logged in to [GitHub](#) with my login credentials.
2. **Created a New Repository:**
  - o Navigate to my GitHub homepage and clicked the "New" button to create a new repository.
  - o Named my repository appropriately, such as Assignment-viewer.
  - o Optional, add a README file, .gitignore, and a license.
3. **Clone the Repository:**
  - o Clone the repository to your local machine using the command:

```
sh
Copy code
git clone https://github.com/username/assignment-viewer.git
```

- o Replace username with your GitHub username.

## Used HTML and CSS to build Assignment Viewer Website

1. **HTML (index.html):**

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>PDF Viewer</title>
  <link rel="stylesheet" href="styles.css">
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&disp
lay=swap" rel="stylesheet">
</head>
<body>
  <header>
    <div class="container">
      <h1>Assignment Viewer</h1>
```

```

        <nav>
            <ul>
                <li><a href="#home">Home</a></li>
                <li><a href="#document">Document</a></li>
                <li><a href="#contact">Contact</a></li>
            </ul>
        </nav>
    </div>
</header>

<main id="home">
    <div class="container">
        <section id="document">
            <h2>Document</h2>
            <div class="pdf-container">
                <embed src="document.pdf" type="application/pdf"
width="100%" height="600px" />
            </div>
        </section>
    </div>
</main>

<footer id="contact">
    <div class="container">
        <p>&copy; 2024 Assignment Viewer. All rights reserved.</p>
    </div>
</footer>
</body>
</html>

```

## 2. CSS (styles.css):

```

css
Copy code
/* Reset some default styles */
body, h1, h2, p, ul, li, a {
    margin: 0;
    padding: 0;
    list-style: none;
    text-decoration: none;
}

/* Body and Container */
body {
    font-family: 'Roboto', sans-serif;
    line-height: 1.6;
    background: #f4f4f4;
    color: #333;
}

.container {
    width: 80%;
    margin: 0 auto;
    overflow: hidden;
    padding: 20px;
}

```

```
/* Header */
header {
    background: #333;
    color: #fff;
    padding: 1rem 0;
    text-align: center;
}

header h1 {
    margin: 0;
    font-size: 2rem;
}

header nav {
    margin-top: 10px;
}

header ul {
    list-style: none;
    padding: 0;
}

header li {
    display: inline;
    margin-left: 15px;
}

header a {
    color: #fff;
    text-decoration: none;
    font-weight: bold;
}

header a:hover {
    color: #ffcc00;
}

/* Main Section */
main {
    padding: 2rem 0;
    background: #eaeaea;
    border-radius: 8px;
    margin-top: 1rem;
}

h2 {
    margin-top: 2rem;
    margin-bottom: 1rem;
    font-size: 1.8rem;
    color: #444;
    text-align: center;
}

.pdf-container {
    margin-bottom: 2rem;
    background: #fff;
}
```

```

border: 1px solid #ccc;
border-radius: 8px;
overflow: hidden;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

/* Footer */
footer {
background: #333;
color: #fff;
text-align: center;
padding: 1rem 0;
margin-top: 2rem;
border-radius: 8px;
}

footer p {
margin: 0;
font-size: 1rem;
}

```

## Deploying the Website on GitHub Pages

### 1. Committed and Pushed My Changes:

- Added my files to the repository:

```
git add .
```

- Committed my changes:

```
git commit -m "Initial commit"
```

- Pushed my changes to GitHub:

```
git push origin main
```

### 2. Enabled GitHub Pages:

- Gone to the repository settings on GitHub.
- Scrolled down to the "GitHub Pages" section.
- Selected the branch you want to use for GitHub Pages, typically the `main` branch, and clicked "Save".
- My site is published at <https://username.github.io/assignment-viewer>.

## Managing and Enhancing My Site

### 1. Regular Updates:

- Keep site updated by regularly committing and pushing new content or changes.

### 2. Custom Domain:

- To use a custom domain, add a CNAME file with your domain name and configure your DNS settings accordingly.

### 3. **SEO and Analytics:**

- Optimize your site for search engines using meta tags and descriptive URLs.
- Use Google Analytics or similar tools to track your site's traffic.

### 4. **Security:**

- Ensure your repository is secure by using strong passwords and enabling two-factor authentication on your GitHub account.

## **Conclusion**

Using GitHub Pages to host your Assignment viewer website is a cost-effective and straightforward method, especially for developers familiar with GitHub. With support for custom domains and the robust version control provided by Git, GitHub Pages is an excellent choice for creating and maintaining static websites.

By following the steps outlined in this report, we can easily set up, deploy, and manage our Assignment viewer website using GitHub Pages, allowing you to share your documents with the world efficiently.