



GitHub Pages Deployment Guide

Complete Step-by-Step Instructions for Deploying Your Portfolio



Prerequisites

- A GitHub account ([Sign up here](https://github.com/join) (<https://github.com/join>) if you don't have one)
 - Git installed on your computer ([Download Git](https://git-scm.com/downloads) (<https://git-scm.com/downloads>))
 - Your portfolio files ready (all files from this project)
-



Step 1: Create GitHub Repository

Option A: User/Organization Site (Recommended)

This will make your site available at `https://username.github.io`

1. **Go to GitHub** and sign in
2. **Click the “+” icon** in the top right corner
3. **Select “New repository”**
4. **Name your repository** exactly as: `username.github.io`
 - Replace `username` with your actual GitHub username
 - Example: If your username is `johnsmith`, name it `johnsmith.github.io`
5. **Set to Public** (required for free GitHub Pages)
6. **Do NOT** initialize with README, .gitignore, or license
7. **Click “Create repository”**

Option B: Project Site

This will make your site available at `https://username.github.io/repository-name`

1. Create a new repository with any name (e.g., `portfolio`, `my-website`)
 2. Set to Public
 3. Click “Create repository”
-



Step 2: Prepare Your Local Files

2.1 Organize Your Files

Ensure your folder structure looks like this:

```

portfolio/
├── index.html
├── style.css
├── script.js
├── assets/
│   ├── images/
│   │   ├── hero-bg.jpg
│   │   ├── about-bg.jpg
│   │   ├── skills-graphic.jpg
│   │   └── projects/
│   │       ├── campus-link.jpg
│   │       ├── safespace-africa.jpg
│   │       ├── kumva-ai.jpg
│   │       ├── survival-analysis.jpg
│   │       ├── mathlearn-chatbot.jpg
│   │       ├── zindi.jpg
│   │       ├── telemed-app.jpg
│   │       └── virtual-assistant.jpg
│   ├── icons/
│   │   └── favicon.svg
│   └── Martin_Koome_CV.pdf # Or your CV filename
├── README.md
└── GITHUB_PAGES_DEPLOYMENT.md

```

2.2 Verify Files

- ☒ `index.html` is in the root folder (NOT in a subfolder)
- ☒ All image paths start with `assets/`
- ☒ CV file is in `assets/`
- ☒ No broken links or missing files

Step 3: Push Files to GitHub

Method 1: Using Git Command Line

1. Open Terminal/Command Prompt

2. Navigate to your portfolio folder:

```

bash
cd path/to/your/portfolio

```

3. Initialize Git repository:

```

bash
git init

```

4. Add all files:

```

bash
git add .

```

5. Commit files:

```

bash
git commit -m "Initial portfolio commit"

```

6. Link to GitHub repository:

```

bash

```

```
git remote add origin https://github.com/username/username.github.io.git
```

Replace `username` with your GitHub username

7. Push to GitHub:

```
bash
```

```
git branch -M main
```


```
git push -u origin main
```

Method 2: Using GitHub Desktop (Easier)

1. **Download and install** [GitHub Desktop](https://desktop.github.com/) (<https://desktop.github.com/>)
2. **Open GitHub Desktop**
3. **Click “Add” → “Add Existing Repository”**
4. **Browse to your portfolio folder**
5. **Click “Create Repository” if prompted**
6. **Write a commit message** (e.g., “Initial portfolio commit”)
7. **Click “Commit to main”**
8. **Click “Publish repository”**
9. **Select your account and repository name**
10. **Uncheck “Keep this code private”** (required for free GitHub Pages)
11. **Click “Publish Repository”**

Method 3: Drag and Drop (Simplest)

1. **Go to your repository on GitHub**
2. **Click “Add file” → “Upload files”**
3. **Drag and drop all your files and folders**
4. **Write a commit message** (e.g., “Initial portfolio commit”)
5. **Click “Commit changes”**

 **Note:** This method can be slow for many files.



Step 4: Enable GitHub Pages

For User/Organization Site (`username.github.io`):

GitHub Pages is **automatically enabled** for repositories named `username.github.io`. Your site will be live at `https://username.github.io` within 1-10 minutes.

For Project Site:

1. **Go to your repository on GitHub**
2. **Click “Settings”** (top right of repository)
3. **Scroll down** and click **“Pages”** in the left sidebar
4. **Under “Source”**, select:
 - Branch: `main` (or `master`)
 - Folder: `/` (root)
5. **Click “Save”**
6. **Wait 1-10 minutes** for deployment
7. **Your site will be live** at `https://username.github.io/repository-name`

✓ Step 5: Verify Deployment

Check Build Status:

1. Go to your repository on GitHub
2. Click the **“Actions”** tab
3. You should see a workflow run named **“pages build and deployment”**
4. Wait for it to complete (green checkmark)
5. If it fails (red X), click on it to see error details

Visit Your Site:

- **User site:** `https://username.github.io`
- **Project site:** `https://username.github.io/repository-name`

Common Issues:

✗ 404 Error:

- Check that `index.html` is in the root directory
- Verify GitHub Pages is enabled in Settings → Pages
- Wait 5-10 minutes and refresh (deployment can be slow)
- Clear browser cache (Ctrl+F5 or Cmd+Shift+R)

✗ Images Not Loading:

- Check image paths in `index.html` and `script.js`
- Ensure paths use forward slashes: `assets/images/hero-bg.jpg`
- Verify files were uploaded to GitHub
- Image paths are case-sensitive on Linux servers

✗ CSS/JS Not Working:

- Check file paths in `index.html`
- Ensure files are in the root directory
- Clear browser cache and hard refresh

🔧 Step 6: Configure Contact Form (EmailJS)

Why EmailJS?

GitHub Pages only supports static sites. EmailJS lets you handle form submissions without a backend server.

Setup Instructions:

1. **Sign up for EmailJS:**
 - Go to <https://www.emailjs.com/> (`https://www.emailjs.com/`)
 - Create a free account (300 emails/month free)
2. **Add Email Service:**
 - Click “Email Services” in the dashboard
 - Click “Add New Service”
 - Select your email provider (Gmail, Outlook, etc.)

- Follow the connection instructions
- Copy your **Service ID**

3. Create Email Template:

- Click “Email Templates” in the dashboard
- Click “Create New Template”
- Set up your template with variables:
`

From: {{from_name}} ({{from_email}})

Subject: {{subject}}

{{message}}

`

- Copy your **Template ID**

4. Get Public Key:

- Click “Account” in the dashboard
- Find your **Public Key** under API Keys

5. Update script.js :

Find lines 370-374 and replace:

```
javascript
const serviceID = 'YOUR_SERVICE_ID';    // Replace with your Service ID
const templateID = 'YOUR_TEMPLATE_ID';  // Replace with your Template ID
const publicKey = 'YOUR_PUBLIC_KEY';    // Replace with your Public Key
```

1. Add EmailJS SDK to index.html :

Before the closing `</body>` tag, add:

```
```html
```

```
```
```

1. Uncomment EmailJS code in script.js :

Find lines 380-387 and uncomment:

```
javascript
const response = await emailjs.send(serviceID, templateID, {
  from_name: formData.name,
  from_email: formData.email,
  subject: formData.subject,
  message: formData.message,
  to_email: 'your-email@domain.com' // Your email
}, publicKey);
```

Remove or comment out the simulated delay line:

```
javascript
// await new Promise(resolve => setTimeout(resolve, 1500)); // Remove this
```

1. Test the form:

- Push changes to GitHub
- Wait for deployment
- Fill out and submit the contact form
- Check your email inbox

Step 7: Customize Your Portfolio

Essential Customizations:

1. **Update Personal Info:**
 - Name, tagline, bio in `index.html`
 - Social media links (LinkedIn, Email)
 - Contact email address
2. **Replace Images:**
 - Add your project images to `assets/images/projects/`
 - Update image paths in `script.js` `projects` array
 - Replace CV file in `assets/`
3. **Update Projects:**
 - Edit the `projects` array in `script.js`
 - Add/remove/modify project details
4. **Customize Colors:**
 - Edit CSS variables in `style.css` (lines 15-30)
5. **Update SEO:**
 - Edit meta tags in `index.html` (lines 5-25)
 - Update title, description, keywords

See [README.md](#) (**README.md**) for detailed customization instructions.

Step 8: Updating Your Site

Whenever you make changes:

Using Git Command Line:

```
git add .
git commit -m "Description of changes"
git push
```

Using GitHub Desktop:

1. Make your changes
2. GitHub Desktop will show changed files
3. Write a commit message
4. Click "Commit to main"
5. Click "Push origin"

Using GitHub Web Interface:

1. Go to your repository
2. Navigate to the file you want to edit
3. Click the pencil icon (Edit)

4. Make your changes
5. Scroll down, write commit message
6. Click “Commit changes”

 **Changes typically go live within 1-5 minutes.**

Step 9: Custom Domain (Optional)

Use your own domain (e.g., `www.yourname.com`) instead of `username.github.io` .

Requirements:

- Own a domain name (purchase from Namecheap, Google Domains, etc.)
- Access to domain DNS settings

Setup Steps:

1. In GitHub:

- Go to repository Settings → Pages
- Under “Custom domain”, enter your domain
- Click “Save”
- Check “Enforce HTTPS” (recommended)

2. In Domain Registrar DNS Settings:

Add these DNS records:

For apex domain (yourname.com):

```

Type: A

Name: @

Value: 185.199.108.153

Type: A

Name: @

Value: 185.199.109.153

Type: A

Name: @

Value: 185.199.110.153

Type: A

Name: @

Value: 185.199.111.153

```

For www subdomain (www.yourname.com):

Type: CNAME

Name: www

Value: username.github.io

1. **Wait for DNS propagation** (can take up to 48 hours, usually < 1 hour)

2. Verify:

- Visit your custom domain
- Should see your portfolio
- HTTPS should work (may take a few hours to activate)

Troubleshooting

Site Not Loading

Check deployment status:

1. Go to repository → Actions tab
2. Look for latest “pages build and deployment”
3. If failed (red X), click to see error

Common causes:

- `index.html` not in root directory
- GitHub Pages not enabled (Settings → Pages)
- Deployment still in progress (wait 5-10 minutes)
- Private repository (must be public for free GitHub Pages)

Images Not Showing

Check paths:

- Paths must be relative: `assets/images/file.jpg`
- NOT absolute: `/assets/images/file.jpg` or `C:/Users/...`
- Case-sensitive: `Hero-bg.jpg` \neq `hero-bg.jpg`

Verify files uploaded:

- Browse your repository on GitHub
- Navigate to `assets/images/`
- Confirm all images are present

Contact Form Not Working

Check EmailJS setup:

1. Verify credentials in `script.js` are correct
2. Check EmailJS script is loaded in `index.html`
3. Test EmailJS service status: status.emailjs.com (<https://status.emailjs.com>)
4. Check browser console (F12) for errors
5. Verify you haven't exceeded free tier limit (300 emails/month)

CSS/JavaScript Not Working

Check file paths in `index.html` :

```
<link rel="stylesheet" href="style.css"> <!-- Correct -->
<script src="script.js"></script> <!-- Correct -->

<!-- NOT: -->
<link rel="stylesheet" href="/style.css"> <!-- Wrong -->
<link rel="stylesheet" href="./style.css"> <!-- Wrong for GitHub Pages -->
```


Clear cache:

- Hard refresh: Ctrl+F5 (Windows) or Cmd+Shift+R (Mac)
- Or use incognito/private mode

404 Error on GitHub Pages

For User Site (`username.github.io`):

- Repository MUST be named exactly `username.github.io`
- `index.html` MUST be in root directory
- Repository MUST be public

For Project Site:

- GitHub Pages must be enabled in Settings
- Source branch must be set correctly
- `index.html` must be in specified folder



Analytics (Optional)

Track visitors to your portfolio:

Google Analytics:

1. **Create account:** analytics.google.com (<https://analytics.google.com>)
2. **Get tracking code**
3. **Add to `index.html` before `</head>` :**

```
```html
```

```
```
```

4. **Push changes to GitHub**



Pre-Deployment Checklist

Before going live, verify:

- ☐ All personal information is updated
- ☐ All project details are accurate
- ☐ All images load correctly
- ☐ CV file is present and downloadable
- ☐ All links work (test every link!)
- ☐ Contact form is configured (EmailJS)
- ☐ SEO meta tags are customized
- ☐ Favicon is set
- ☐ Mobile responsive (test on phone)
- ☐ No console errors (open DevTools with F12)
- ☐ Tested in multiple browsers
- ☐ Social media links are correct
- ☐ Spelling and grammar checked

You're Live!

Congratulations! Your portfolio is now live on the internet.


Next Steps:

1. **Share your portfolio:**
 - Add to LinkedIn profile
 - Include in email signature
 - Share on social media
 - Add to resume/CV
 2. **Promote your portfolio:**
 - Submit to portfolio showcases
 - Share in relevant communities
 - Include in job applications
 3. **Keep it updated:**
 - Add new projects regularly
 - Update skills as you learn
 - Refresh project descriptions
 - Update CV periodically
 4. **Monitor performance:**
 - Check analytics (if enabled)
 - Test on different devices
 - Get feedback from peers
 - Iterate and improve
-

Need Help?

If you're stuck:

1. **Re-read this guide carefully**
 2. **Check the [README.md](#) (README.md) troubleshooting section**
 3. **Search Google/Stack Overflow for specific errors**
 4. **Check GitHub Pages documentation: docs.github.com/pages (<https://docs.github.com/pages>)**
 5. **Ask in GitHub Discussions or relevant communities**
-

Good luck with your portfolio! 

You've got this! 