

Complete Guide: Deploying React Vite Application to GitHub Pages

Table of Contents

1. [Prerequisites](#)
 2. [Project Setup](#)
 3. [GitHub Repository Setup](#)
 4. [Vite Configuration](#)
 5. [GitHub Actions Workflow](#)
 6. [Deployment Process](#)
 7. [Troubleshooting](#)
 8. [Verification](#)
-

Prerequisites

Before you begin, ensure you have:

- ☒ Node.js (v18 or higher) installed
 - ☒ Git installed and configured
 - ☒ GitHub account
 - ☒ React Vite project ready to deploy
-

Project Setup

Step 1: Verify Your Vite Project Structure

```
my-app/  
├── public/  
├── src/  
├── index.html  
├── package.json  
├── vite.config.js  
└── .gitignore
```

Step 2: Install Required Dependencies

```
cd my-app  
npm install
```

Step 3: Test Local Build

```
npm run build
npm run preview
```

- ☒ Verify the build works correctly before proceeding.

GitHub Repository Setup

Step 1: Choose Repository Type

Option A: User/Organization Site (Recommended for portfolios)

- Repository name: `<username>.github.io`
- Example: `MuthuGanesh-dev.github.io`
- Deployment URL: `https://muthuganesh-dev.github.io/`
- **Base path:** `/` (root)

Option B: Project Site

- Repository name: Any name (e.g., `my-portfolio`)
- Example: `MuthuGanesh-dev/my-portfolio`
- Deployment URL: `https://muthuganesh-dev.github.io/my-portfolio/`
- **Base path:** `/my-portfolio/`

Step 2: Create GitHub Repository

1. Go to `https://github.com/new`
2. Enter repository name (following Option A or B above)
3. Set to **Public** (required for free GitHub Pages)
4. Don't initialize with README
5. Click **Create repository**

Step 3: Initialize Local Git Repository

```
# Navigate to your project root (parent of my-app/)
cd d:\All_React_app\Muthuganesh_portfolio

# Initialize git if not already done
git init

# Add remote origin
git remote add origin https://github.com/<username>/<repository-name>.git

# Example for user site:
git remote add origin https://github.com/MuthuGanesh-dev/MuthuGanesh-dev.github.io.git
```

⚙️ Vite Configuration

Step 1: Update `vite.config.js`

Open `my-app/vite.config.js` and configure the `base` path:

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

```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react-swc";
import tailwindcss from "@tailwindcss/vite";
import path from "path";

export default defineConfig({
  plugins: [react(), tailwindcss()],
  base: "/", // ★ ROOT path for username.github.io repos
  resolve: {
    alias: {
      "@": path.resolve(__dirname, "./src"),
    },
  },
  server: {
    proxy: {
      "/api": {
        target: "http://localhost:3001",
        changeOrigin: true,
      },
    },
  },
});
```

For Project Site (`<username>.github.io/<repo-name>`):

```
export default defineConfig({
  plugins: [react(), tailwindcss()],
  base: "/my-portfolio/", // ★ SUBFOLDER path for project repos
  // ... rest of config
});
```

Step 2: Verify Build Output

```
cd my-app
npm run build
```

Check that `my-app/dist/` folder is created with all assets.

GitHub Actions Workflow

Step 1: Create Workflow Directory

```
# From project root
mkdir -p .github/workflows
```

Step 2: Create Workflow File

Create `.github/workflows/deploy.yml`:

```
name: Deploy to GitHub Pages

on:
  # Runs on pushes to main branch
  push:
    branches: ["main"]

  # Allows manual trigger from Actions tab
  workflow_dispatch:

# Required permissions for GitHub Pages deployment
permissions:
  contents: read
  pages: write
  id-token: write

# Allow one concurrent deployment
concurrency:
  group: "pages"
  cancel-in-progress: true

jobs:
  # Build job
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: "20"
          cache: "npm"
          cache-dependency-path: "./my-app/package-lock.json"

      - name: Install dependencies
        working-directory: ./my-app
        run: npm ci
```

```
- name: Build application
  working-directory: ./my-app
  run: npm run build

- name: Setup Pages
  uses: actions/configure-pages@v4

- name: Upload artifact
  uses: actions/upload-pages-artifact@v3
  with:
    path: "./my-app/dist"

# Deployment job
deploy:
  environment:
    name: github-pages
    url: ${ steps.deployment.outputs.page_url }
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4
```

Step 3: Understand the Workflow

Triggers:

- Automatically runs when you push to **main** branch
- Can be manually triggered from Actions tab

Build Job:

1. Checks out your code
2. Sets up Node.js v20
3. Installs dependencies with **npm ci** (clean install)
4. Runs **npm run build** in **my-app/** directory
5. Uploads the **my-app/dist/** folder as artifact

Deploy Job:

1. Takes the artifact from build job
2. Deploys it to GitHub Pages
3. Provides deployment URL

Deployment Process

Step 1: Commit Your Code

```
# From project root
git add .
git commit -m "Initial commit: Setup portfolio with GitHub Pages deployment"
```

Step 2: Push to GitHub

```
git branch -M main
git push -u origin main
```

Step 3: Enable GitHub Pages

1. Go to your repository on GitHub
2. Click **Settings** tab
3. Scroll to **Pages** section (left sidebar)
4. Under **Build and deployment**:
 - **Source**: Select **GitHub Actions** (not "Deploy from a branch")
5. Click **Save**

Screenshot Guide:

Settings → Pages → Build and deployment

Source: [GitHub Actions ▼]

← Select this

✗ Not "Deploy from a branch"

Step 4: Monitor Deployment

1. Go to **Actions** tab in your repository
2. You should see "Deploy to GitHub Pages" workflow running
3. Click on the workflow to see detailed logs
4. Wait for both **build** and **deploy** jobs to complete (green checkmarks ☒)

Typical Timeline:

- Build: 2-3 minutes
- Deploy: 1 minute
- **Total: ~3-5 minutes**

Verification

Step 1: Check Deployment Status

In Actions Tab:

- ☒ Green checkmark on workflow
- ☒ "Deploy to GitHub Pages" successful
- Click on workflow → Click "deploy" job → See deployment URL

Step 2: Access Your Site**For User Site:**

- URL: <https://<username>.github.io/>
- Example: <https://muthuganesh-dev.github.io/>

For Project Site:

- URL: <https://<username>.github.io/<repo-name>/>
- Example: <https://muthuganesh-dev.github.io/my-portfolio/>

Step 3: Test Functionality☒ Check the following:

- ☐ Page loads without errors
- ☐ All images load correctly
- ☐ Videos play (if applicable)
- ☐ Theme toggle works (light/dark mode)
- ☐ Navigation links work
- ☐ External links (GitHub, LinkedIn) work
- ☐ Mobile responsiveness

Open Browser Console (F12):

- Should see no errors in Console tab
- All assets should load (Network tab)

Troubleshooting

Issue 1: MIME Type Error

```
Failed to load module script: Expected a JavaScript module
but the server responded with a MIME type of "text/html"
```

Solution: Incorrect `base` path in `vite.config.js`

- For `<username>.github.io` → Use `base: "/"`
- For `<username>.github.io/<repo>` → Use `base: "/<repo>/"`

Fix:

```
// vite.config.js
export default defineConfig({
  base: "/", // Change this based on your repo type
  // ...
});
```

Then rebuild and push:

```
cd my-app
npm run build
git add .
git commit -m "Fix base path for deployment"
git push
```

Issue 2: 404 Page Not Found

Cause: GitHub Pages not enabled or wrong source selected.

Solution:

1. Go to Settings → Pages
2. Ensure **Source** is set to **GitHub Actions** (not branch)
3. Wait 1-2 minutes for changes to propagate

Issue 3: Workflow Fails at Build Step

Check `package-lock.json` exists:

```
cd my-app
ls package-lock.json # Should exist
```

If missing, generate it:

```
npm install
git add package-lock.json
git commit -m "Add package-lock.json"
git push
```

Issue 4: Assets Not Loading (Images/Videos)

Check file paths in code:

```
// ☒ Correct (relative to public/)

```



```
// ❌ Wrong  
  

```

Verify files in `public/` directory:

```
my-app/public/  
├─ images/  
│   └─ profile.jpg  
└─ videos/  
    └─ demo.mp4
```

Issue 5: Deployment Takes Too Long**Check workflow logs:**

- Actions tab → Click workflow → Expand each step
- Look for errors in "Install dependencies" or "Build application"

Common fixes:

```
# Clear node_modules and reinstall  
cd my-app  
rm -rf node_modules package-lock.json  
npm install  
git add package-lock.json  
git commit -m "Regenerate package-lock.json"  
git push
```

Issue 6: Changes Not Reflecting on Site**Browser cache issue:**

- Hard refresh: `Ctrl + Shift + R` (Windows) or `Cmd + Shift + R` (Mac)
- Clear browser cache
- Try incognito/private window

Deployment delay:

- Wait 2-3 minutes after workflow completes
- Check Actions tab for successful deployment

Issue 7: GitHub Actions Not Triggering**Verify workflow file location:**

```
.github/  
└─ workflows/  
    └─ deploy.yml # Must be exactly this path
```

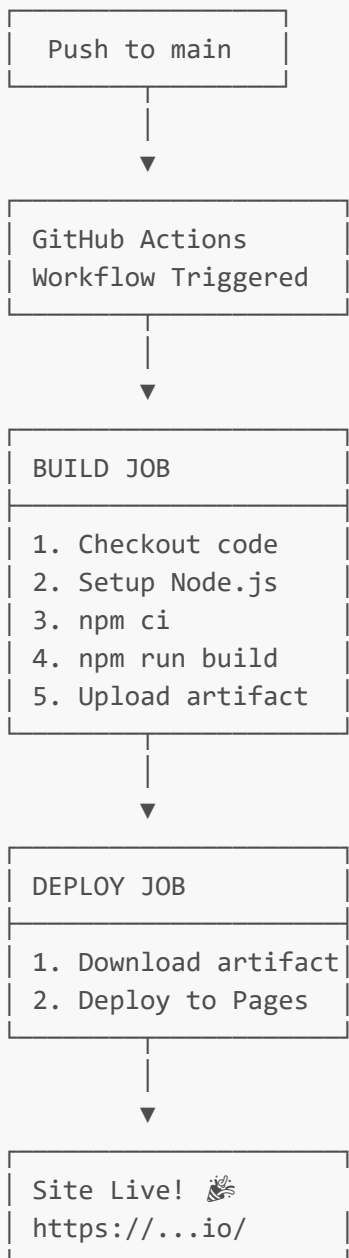
Check file syntax:

- YAML is indentation-sensitive (use 2 spaces, not tabs)
- Validate at: <https://www.yamllint.com/>

Manual trigger:

- Actions tab → "Deploy to GitHub Pages" → "Run workflow" → Run

Workflow Diagram



Complete Deployment Checklist

Pre-Deployment

- ☐ Vite project builds locally (`npm run build`)
- ☐ Preview works (`npm run preview`)
- ☐ GitHub repository created
- ☐ Repository is **Public**

Configuration

- ☐ `vite.config.js` has correct `base` path
- ☐ `.github/workflows/deploy.yml` created
- ☐ Workflow has correct `working-directory` path
- ☐ `package-lock.json` exists in project

Git Setup

- ☐ Git initialized (`git init`)
- ☐ Remote added (`git remote add origin ...`)
- ☐ All files committed
- ☐ Pushed to `main` branch

GitHub Settings

- ☐ GitHub Pages enabled
- ☐ Source set to **GitHub Actions**
- ☐ Repository Settings → Pages configured

Post-Deployment

- ☐ Workflow completed successfully (green ☒)
- ☐ Site accessible at deployment URL
- ☐ No console errors in browser
- ☐ All features working
- ☐ Mobile responsive

Quick Reference Commands

```
# Navigate to project
cd my-app

# Install dependencies
npm install

# Test build
npm run build
```

```
# Preview build
npm run preview

# Git operations
git add .
git commit -m "Deploy to GitHub Pages"
git push origin main

# Check git status
git status

# View git log
git log --oneline

# Check remote
git remote -v
```

Useful Resources

- [Vite Documentation](#)
- [GitHub Pages Documentation](#)
- [GitHub Actions Documentation](#)
- [Deploy Pages Action](#)

Pro Tips

1. **Use User Site for Portfolios:** `<username>.github.io` gives cleaner URLs
2. **Test Locally First:** Always run `npm run build && npm run preview` before deploying
3. **Monitor Actions:** Check Actions tab after every push to catch issues early
4. **Environment Variables:** Use `.env` files for API keys (don't commit them!)
5. **Custom Domain:** You can add custom domain in Settings → Pages
6. **HTTPS:** GitHub Pages automatically provides HTTPS
7. **Branch Protection:** Protect `main` branch to prevent accidental force pushes

Success!

Your React Vite application is now live on GitHub Pages!

Next Steps:

- Share your portfolio URL
- Add custom domain (optional)
- Set up analytics (Google Analytics, etc.)
- Add SEO meta tags
- Configure robots.txt and sitemap.xml

Maintenance:

- Every push to **main** automatically deploys
 - No manual build/upload needed
 - Free hosting forever (for public repos)
-

Created: November 11, 2025

Author: GitHub Copilot

Project: MuthuGanesh Portfolio

Repository: <https://github.com/MuthuGanesh-dev/MuthuGanesh-dev.github.io>