🚀 GitHub as Backend - Complete Setup Guide

📖 Overview

This guide explains how to use **GitHub repository as a database** for your portfolio website. Instead of paying for database hosting, your projects are stored directly in your GitHub repo's `projects.json` file using the GitHub API.

🎯 How It Works

```
User adds/deletes project
       ↓
[Password Authentication]
       ↓
[Save to localStorage] (Instant - this device only)
       ↓
[GitHub API Call] (Commits to projects.json)
       ↓
[GitHub Pages Auto-Rebuild] (~2-3 minutes)
       ↓
[Live on Website] 🎉
       ↓
All users worldwide see the update!
```

📋 Prerequisites

Before starting, you need:

- ☑ GitHub account
- ☑ Repository for your portfolio (MuthuGanesh-dev.github.io)
- ☑ Collaborator access (if working on someone else's repo)
- ☑ Node.js and npm installed

🔧 Step-by-Step Setup

Step 1: Generate GitHub Personal Access Token (PAT)

This token allows your portfolio to make changes to the repository.

For Classic Token (Recommended):

1. Go to: https://github.com/settings/tokens
2. Click **"Generate new token"** → **"Generate new token (classic)"**
3. Configure the token:

- **Note**: `Portfolio Management Token`
- **Expiration**: `No expiration` (or choose a custom period)
- **Scopes**: Check ☑ `repo` (Full control of private repositories)

4. Click **"Generate token"** at the bottom
5. **IMPORTANT**: Copy the token immediately! It looks like:
   `ghp_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
6. Store it safely - you won't see it again!

For Fine-Grained Token (More Secure):

1. Go to: https://github.com/settings/tokens?type=beta
2. Click **"Generate new token"**
3. Configure:
   - **Token name**: `Portfolio Management`
   - **Expiration**: Choose a period
   - **Repository access**: Select "Only select repositories"
   - Choose: `MuthuGanesh-dev.github.io`
   - **Permissions**:
     - Contents: **Read and write** ☑
4. Click **"Generate token"**
5. Copy the token (looks like: `github_pat_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`)

Step 2: Set Up Local Development Environment

2.1: Create `.env` File

In your project folder `my-app/`, create a file named `.env`:

```
cd my-app
```

Create `.env` file with this content:

```
# GitHub Personal Access Token for project management
# Create at: https://github.com/settings/tokens
# Required scope: repo (Full control of private repositories)
VITE_GITHUB_TOKEN=ghp_YOUR_ACTUAL_TOKEN_HERE

# Admin password for add/delete authentication
ADMIN_PASSWORD=ganesh3012

# API Base URL (optional, leave blank for GitHub storage)
# Only needed if using external API instead of GitHub
VITE_API_BASE_URL=
```

Replace `ghp_YOUR_ACTUAL_TOKEN_HERE` with your actual token from Step 1!

2.2: Verify `.gitignore` Protection

Make sure `.env` is listed in `my-app/.gitignore` to prevent committing secrets:

```
# Environment variables
.env
.env.local
.env.production.local
.env.development.local
```

☑ This is already set up in your project!

Step 3: Configure GitHub Storage Module

The file `my-app/src/utils/githubStorage.js` contains the GitHub API integration.

Key Configuration (already set up):

```
const GITHUB_OWNER = "MuthuGanesh-dev";
const GITHUB_REPO = "MuthuGanesh-dev.github.io";
const GITHUB_FILE_PATH = "my-app/public/projects.json";
const GITHUB_BRANCH = "main";
```

This module provides two functions:

- `saveProjectsToGitHub(projects, password)`: Commits changes to GitHub
- `loadProjectsFromGitHub()`: Fetches latest projects from GitHub

Step 4: Test Locally

4.1: Install Dependencies

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

4.2: Start Development Server

```
npm run dev
```

Server will start at: http://localhost:5173 or http://localhost:5174

4.3: Test Adding a Project

1. Open http://localhost:5174 in your browser
2. Click **"Add Project"** button
3. Enter password: `ganesh3012` (or your custom password)
4. Fill in project details:
    - **Title**: Test Project
    - **Description**: Testing GitHub storage
    - **Technologies**: React, Vite (comma-separated)
    - **Video URL**: (optional)
    - **PDF URL**: (optional)
    - **Link**: https://example.com
5. Click **"Add Project"**
6. You should see: ☑ **"Projects saved to GitHub!"**

4.4: Verify GitHub Commit

1. Go to: https://github.com/MuthuGanesh-dev/MuthuGanesh-dev.github.io/commits/main
2. You should see a new commit: **"Update projects - [timestamp]"**
3. Click the commit to see changes to `my-app/public/projects.json`

4.5: Test Persistence

1. **Reload the page** (F5 or Ctrl+R)
2. Your project should **still be there**! 🎉
3. This confirms localStorage + GitHub sync is working

Step 5: Set Up Production (GitHub Pages)

For your live website to save projects, you need to add the token as a GitHub Secret.

5.1: Add GitHub Secret

1. Go to your repository settings:
    - https://github.com/MuthuGanesh-dev/MuthuGanesh-dev.github.io/settings/secrets/actions
2. Click **"New repository secret"**
3. Fill in:
    - **Name**: `VITE_GITHUB_TOKEN`
    - **Secret**: Paste your GitHub Personal Access Token
4. Click **"Add secret"**

5.2: Verify Workflow Configuration

The file `.github/workflows/deploy.yml` should have this in the Build step:

```
- name: Build
  env:
    VITE_GITHUB_TOKEN: ${{ secrets.VITE_GITHUB_TOKEN }}
  run: |
    cd my-app
    npm run build
```

☑ This is already configured!

5.3: Trigger Deployment

Push any change to trigger GitHub Actions:

```
git add .
git commit -m "Enable GitHub backend for production"
git push origin main
```

5.4: Monitor Deployment

1. Go to: https://github.com/MuthuGanesh-dev/MuthuGanesh-dev.github.io/actions
2. Click on the latest workflow run
3. Wait for both **"build"** and **"deploy"** jobs to complete (green checkmark ☑)
4. Your site will be live at: https://muthuganesh-dev.github.io/

🔒 Security Best Practices

☑ DO:

- ☑ Keep `.env` file in `.gitignore`
- ☑ Use GitHub Secrets for production tokens
- ☑ Set token expiration dates (fine-grained tokens)
- ☑ Use fine-grained tokens with minimal permissions
- ☑ Change admin password from default `ganesh3012`
- ☑ Regenerate tokens if accidentally exposed

❌ DON'T:

- ❌ Never commit `.env` file to git
- ❌ Never share your token publicly
- ❌ Never hardcode tokens in source code
- ❌ Never push `.env` to GitHub
- ❌ Never use tokens in client-side code for sensitive operations

---

🛠 Troubleshooting

Problem 1: "GitHub token missing" Error

Cause: `.env` file doesn't have a valid token

Solution:

1. Check `my-app/.env` file exists
2. Verify token starts with `ghp_` or `github_pat_`
3. Make sure no extra spaces or quotes around token
4. Restart dev server: `npm run dev`

Problem 2: "Failed to fetch current file" Error

Cause: Token doesn't have correct permissions

Solution:

1. Go to https://github.com/settings/tokens
2. Click on your token
3. For classic: Ensure `repo` scope is checked
4. For fine-grained: Ensure **Contents: Read and write** is enabled
5. Regenerate token if needed

Problem 3: Projects Disappear on Reload

Cause: GitHub storage not integrated or token missing

Solution:

1. Check browser console for errors (F12)
2. Verify `.env` has valid token
3. Check `Portfolio.jsx` imports `githubStorage` functions
4. Verify `handleAddProject` calls `saveProjectsToGitHub()`

Problem 4: Production Site Shows Error

Cause: GitHub Secret not configured

Solution:

1. Go to repo Settings → Secrets and variables → Actions
2. Verify `VITE_GITHUB_TOKEN` secret exists
3. Re-add the secret if needed
4. Push a new commit to trigger rebuild

Problem 5: "403 Forbidden" from GitHub API

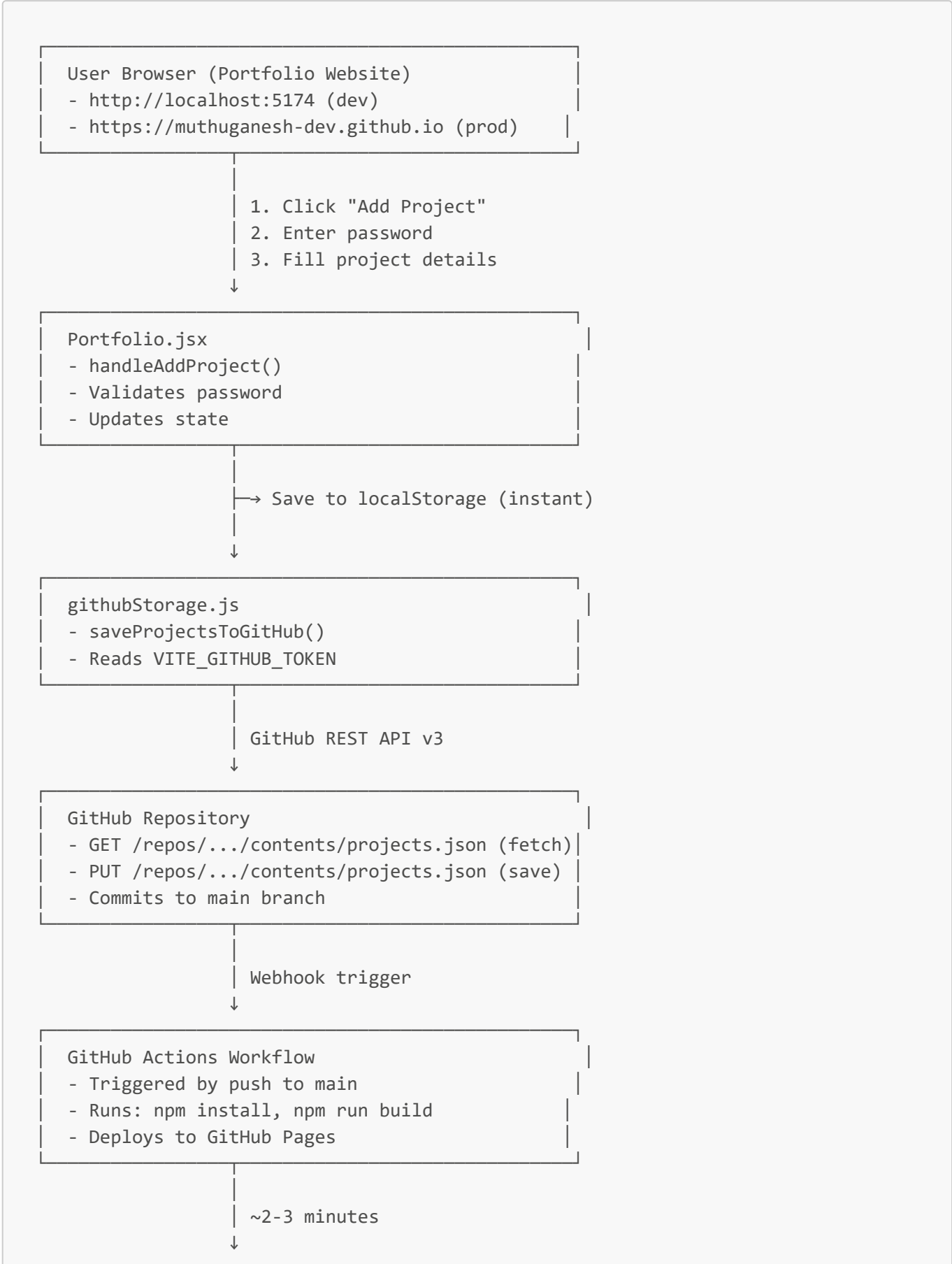
Cause: Token expired or lacks permissions

Solution:

1. Check token expiration date

2. Regenerate token with correct scopes
3. Update `.env` and GitHub Secret with new token

Architecture Diagram



```
┌──────────────────────────────────────────────┐
│  Live Website Updated! 🎉                    │
│  https://muthuganesh-dev.github.io/          │
│  All users see new project                   │
└──────────────────────────────────────────────┘
```

File Structure

```
MuthuGanesh-dev.github.io/
├── .github/
│   └── workflows/
│       └── deploy.yml          # GitHub Actions workflow (with VITE_GITHUB_TOKEN)
├── my-app/
│   ├── .env                    # Local env variables (NOT committed)
│   ├── .env.example            # Template (safe to commit)
│   ├── .gitignore              # Protects .env from being committed
│   ├── public/
│   │   └── projects.json       # Project data (modified by GitHub API)
│   ├── src/
│   │   ├── pages/
│   │   │   └── Portfolio.jsx   # Main portfolio component
│   │   └── utils/
│   │       └── githubStorage.js # GitHub API integration
│   └── package.json
└── GITHUB_BACKEND_SETUP.md     # This file
```

Key Concepts

What is GitHub as a Backend?

Instead of using traditional databases (MongoDB, PostgreSQL, etc.), your portfolio uses:

- **GitHub repository** as the database
- **projects.json** as the data file
- **GitHub API** as the data access layer
- **GitHub Actions** as the deployment pipeline

Why Use GitHub as Backend?

Advantages:

- ☑ **100% Free** - No hosting costs
- ☑ **Global CDN** - GitHub Pages is fast worldwide
- ☑ **Version Control** - Full history of all changes
- ☑ **Auto Deployment** - Changes go live automatically
- ☑ **Simple Setup** - No server configuration needed
- ☑ **Reliable** - GitHub's 99.9% uptime

Trade-offs:

- ⚠ **2-3 minute delay** for changes to go live (GitHub Pages rebuild time)
- ⚠ **Public data** - Everything in repo is visible (fine for portfolios)
- ⚠ **Rate limits** - GitHub API has limits (5000 requests/hour, enough for portfolios)
- ⚠ **Not for sensitive data** - Don't store private information

---

🎓 Learning Resources

- **GitHub REST API**: https://docs.github.com/en/rest
- **GitHub Actions**: https://docs.github.com/en/actions
- **Vite Environment Variables**: https://vite.dev/guide/env-and-mode.html
- **GitHub Pages**: https://docs.github.com/en/pages

☑ Checklist

Before going live, verify:

- ☐ GitHub Personal Access Token created
- ☐ `.env` file created with valid token
- ☐ `.env` is in `.gitignore`
- ☐ Local testing successful (add/delete/reload works)
- ☐ GitHub Secret `VITE_GITHUB_TOKEN` added to repo
- ☐ Workflow file has `env:` section in Build step
- ☐ Test project committed to GitHub successfully
- ☐ GitHub Actions workflow runs without errors
- ☐ Live website loads at https://muthuganesh-dev.github.io/
- ☐ Can add/delete projects on live site
- ☐ Changes persist after page reload

🎉 Congratulations!

You now have a fully functional portfolio with:

- ☑ Persistent storage (no data loss on reload)
- ☑ Global sync (all devices see same data)
- ☑ Free hosting (zero costs)
- ☑ Password protection (secure management)
- ☑ Automatic deployment (push and forget)

Your portfolio is production-ready! 🚀

📞 Need Help?

If you encounter issues:

1. **Check browser console** (F12) for error messages
2. **Check GitHub Actions logs** for deployment errors
3. **Verify `.env` file** has correct token format
4. **Review this guide** for missed steps
5. **Check GitHub API status**: https://www.githubstatus.com/

Last Updated: November 13, 2025
Version: 1.0
Author: GitHub Copilot