

Quick Setup Guide

Technical setup reference for deploying and configuring Vexel Logic.

■ Deploy to Vercel (5 Minutes)

Step 1: Sign Up for Vercel

1. Visit: <https://vercel.com>
2. Click "Sign Up"
3. Choose "Continue with GitHub"
4. Authorize Vercel to access your repositories

Step 2: Import Your Project

1. Click "Add New..." → "Project"
2. Select "Import Git Repository"
3. Find your `vexel-logic` repository
4. Click "Import"

Step 3: Configure Deployment

Framework Preset: Other (static site)

Root Directory: `./`

Build Command: (leave blank)

Output Directory: (leave blank)

Click **"Deploy"**

Step 4: Wait & Access

- Deployment takes 1-2 minutes

- You'll get a URL like: `vexel-logic.vercel.app`
- Click the URL to view your live site

Done! Your site is live.

Auto-Deploy on Every Push

Every time you push to GitHub:

1. Vercel automatically detects the change
2. Builds and deploys in ~1 minute
3. Your site updates automatically

No manual steps needed!

■ Connect Calendly (15 Minutes)

Step 1: Create Calendly Account

1. Go to: <https://calendly.com>
2. Sign up (free plan works)
3. Create event type:
 - Name: "Vexel Logic Demo"
 - Duration: 30 minutes
 - Location: Zoom or Google Meet

Step 2: Get Your Link

Your link looks like: `https://calendly.com/your-username/vexel-demo`

Step 3: Update Website

1. Open `index.html`
2. Find (Ctrl+F): `https://calendly.com/vexellogic/demo`
3. Replace with your actual Calendly link

4. Save file

Step 4: Push Changes

```
git add index.html  
git commit -m "Update Calendly link"  
git push
```

Done! Demo bookings will go to your calendar.

■ Add Google Analytics (10 Minutes)

Step 1: Create GA4 Property

1. Go to: <https://analytics.google.com>
2. Create account → Create property
3. Property name: "Vexel Logic"
4. Copy your Measurement ID (looks like: `G-XXXXXXXXXXX`)

Step 2: Add to Website

1. Open `index.html`
2. Add this code inside `` tag:

```
<!-- Google Analytics -->  
<script async src="https://www.googletagmanager.com/gtag/js?id=G-XXXXXXXXXXX"></script>  
<script>  
  window.dataLayer = window.dataLayer || [];  
  function gtag(){dataLayer.push(arguments);}  
  gtag('js', new Date());  
  gtag('config', 'G-XXXXXXXXXXX');  
</script>
```

3. Replace `G-XXXXXXXXXXX` with your actual ID

Step 3: Deploy

```
git add index.html
```

```
git commit -m "Add Google Analytics"  
git push
```

Step 4: Verify

1. Visit your live site
2. Check Google Analytics → Real-time report
3. You should see yourself as active user

Done! Now tracking visitors.

■■ Local Development Setup

Method 1: Python Server (Recommended)

```
cd "C:\Users\Nxiss\OneDrive\Desktop\Vexel project"  
python -m http.server 8000
```

Open: `http://localhost:8000`

Stop server: Ctrl+C

Method 2: VS Code Live Server

1. Install extension: "Live Server" by Ritwick Dey
2. Right-click `index.html`
3. Select "Open with Live Server"
4. Auto-opens at `http://127.0.0.1:5500`

Bonus: Auto-refreshes when you save files!

Method 3: Double-Click

Just double-click `index.html`

Note: Some features may not work (CORS issues)

■ Custom Domain (Optional)

Step 1: Buy Domain

Recommended registrars:

- Namecheap.com (£10/year)
- GoDaddy.com
- Google Domains

Example: `vexellogic.com`

Step 2: Configure GitHub Pages

1. Go to: <https://github.com/Nxiss12/vexel-logic/settings/pages>

2. Under "Custom domain":

- Enter your domain: `vexellogic.com`
- Click **Save**

Step 3: Update DNS

Add these records at your registrar:

For apex domain (vexellogic.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.vexellogic.com):

```
Type: CNAME  
Name: www  
Value: nxiss12.github.io
```

Step 4: Wait & Verify

- DNS propagation: 1-48 hours (usually 1-4 hours)
- Check: <https://dnschecker.org>
- SSL auto-enabled after DNS resolves

Done! Your site is at your custom domain.

■ Payment Setup (Stripe)

Step 1: Create Stripe Account

1. Go to: <https://stripe.com>
2. Sign up
3. Verify business info
4. Get API keys

Step 2: Add to Website

Coming soon: Payment integration guide

For now: Use manual invoicing (see below)

■ Email Setup

Option 1: Gmail (Free)

Use: `your-name@gmail.com`

Pros: Free, easy

Cons: Looks less professional

Option 2: Custom Email (Recommended)

Providers:

- Google Workspace (£4.60/month)
- Microsoft 365 (£3.80/month)
- Zoho Mail (£0.80/month)

Example: `hello@vexellogic.com`

Setup: Follow provider's instructions (usually automatic)

■■ Tool Configuration

Update Supabase Keys

If using backend database:

1. Create Supabase account: <https://supabase.com>
2. Create new project
3. Get API keys
4. Open `index.html`
5. Find: `supabaseUrl` and `supabaseKey`
6. Replace with your actual keys

Note: For now, tools work without backend (static mode)

■ Testing Checklist

Before going live, test:

- [] Homepage loads correctly
- [] All styles display (Tailwind CSS)
- [] Calculator works
- [] Tools page opens: `/tools/index.html`
- [] Marketing tools open: `/tools/marketing_tools/index.html`
- [] Calendly button works
- [] Mobile responsive (test on phone or Chrome DevTools)
- [] No console errors (F12 → Console)
- [] Forms submit (if using Supabase)

■ Troubleshooting

"Page build failed"

Check:

- `index.html` is in root folder
- GitHub Actions tab for error details
- Wait 5 minutes and try again

"404 Not Found"

Fix:

- Add `/` at end of URL
- Clear cache: Ctrl+Shift+R
- Wait 3-5 minutes after deployment

"Styles not loading"

****Fix:****

- Check internet connection (CDN required)
- Browser console for errors
- Try different browser

"Can't push to GitHub"

****Fix:****

```
# Check status  
git status  
  
# Pull latest changes  
git pull origin main  
  
# Then push again  
git push
```

"Python not found"

****Fix:****

1. Download: <https://www.python.org/downloads/>
2. During install: Check "Add Python to PATH"
3. Restart terminal
4. Try again

■ Important Files

File	Purpose
-----	-----

`index.html`	Main website
`tools/`	All tools
`START-HERE.md`	Your entry point
`.gitignore`	Files to ignore in Git

■ Updating Your Site

Make changes:

1. Edit files locally
2. Test locally (Python server)
3. Commit changes:

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

4. Wait 2-3 minutes
5. Check live site

■ Pro Tips

1. **Test locally first** - Don't push broken code
2. **Commit often** - Small commits are easier to track
3. **Write good commit messages** - Future you will thank you
4. **Keep backups** - Git is already a backup, but export database
5. **Monitor uptime** - Use UptimeRobot (free)

■ Additional Resources

Full hosting guide:/STRATEGY-VAULT/docs/FREE_HOSTING_SETUP.md

****Local development:****/STRATEGY-VAULT/docs/LOCAL_DEVELOPMENT.md

****Need help?****

- Email: benanokye577@gmail.com
- Check browser console (F12) for errors

****Setup complete?**** Go back to: Week-1-Actions.md