Setup Automation Script - New Client Deployment

Follow these steps to deploy an isolated website for each new client.

Time estimate: 25-30 minutes per client

PREREQUISITES

Before starting, ensure you have:

- [ ] Client onboarding form completed
- [ ] All photos and logo files downloaded
- [ ] Supabase account (https://supabase.com)
- [ ] Twilio account (https://twilio.com)
- [ ] Vercel account (https://vercel.com)
- [ ] GitHub repository with template

CLIENT ISOLATION STRATEGY

Each client gets:

1. **Separate Supabase project** (their own database, completely isolated)
2. **Unique Twilio phone number** (their tracking number only)
3. **Own Vercel deployment** (separate app instance)
4. **Own domain** (custom domain or subdomain)

Why isolation matters:

- Client A's data never touches Client B's data
- If one database fails, others unaffected
- Easy to export client data if they leave
- Simpler security (each project has own credentials)

STEP 1: CREATE SUPABASE PROJECT (5 minutes)

1.1 Create New Project

1. Go to https://app.supabase.com
2. Click **"New Project"**
3. Fill in:
   - **Name:** `[client-slug]-website`
     - Example: `atlanta-hvac-website`
   - **Database Password:** Generate strong password (save this!)
   - **Region:** `East US (North Virginia)` (or closest to client)
   - **Pricing Plan:** Free tier (upgrade to Pro at $25/month when ready)
4. Click **"Create new project"**
5. Wait 2-3 minutes for provisioning

1.2 Save Connection Details

Once project is ready:

1. Go to **Project Settings → API**

Copy and save these to `PROJECT_SECRETS_REFERENCE.txt`:
Client: [Business Name]Supabase Project: [client-slug]-websiteSupabase URL: https://xxxxx.supabase.coSupabase Anon Key: eyJhbGc...Supabase Service Role Key: eyJhbGc...Database Password: [saved from step 1.1]

2.

1.3 Create Database Tables

1. Go to **SQL Editor**
2. Click **"New query"**
3. Copy/paste this SQL and run:

```
-- Table 1: Leads (form submissions)
CREATE TABLE leads (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

  full_name TEXT,
  phone TEXT,
  email TEXT,
  message TEXT,
  source TEXT DEFAULT 'form',
  status TEXT DEFAULT 'new',

  ip_address TEXT,
  user_agent TEXT
);

CREATE INDEX idx_leads_created ON leads(created_at);
CREATE INDEX idx_leads_status ON leads(status);

-- Table 2: Call Logs (Twilio calls)
CREATE TABLE call_logs (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

  from_number TEXT NOT NULL,
  to_number TEXT NOT NULL,
  call_sid TEXT UNIQUE,
  duration INTEGER,
  status TEXT,
  recording_url TEXT,
  call_direction TEXT
);

CREATE INDEX idx_call_logs_created ON call_logs(created_at);

-- Table 3: SMS Logs (Twilio SMS)
CREATE TABLE sms_logs (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

  from_number TEXT NOT NULL,
  to_number TEXT NOT NULL,
  message_body TEXT,
  message_sid TEXT UNIQUE,
  status TEXT,
  direction TEXT,
  triggered_by TEXT
);

CREATE INDEX idx_sms_logs_created ON sms_logs(created_at);

-- Enable Row Level Security (for future dashboard)
ALTER TABLE leads ENABLE ROW LEVEL SECURITY;
ALTER TABLE call_logs ENABLE ROW LEVEL SECURITY;
ALTER TABLE sms_logs ENABLE ROW LEVEL SECURITY;
```

4. Click **"Run"**
5. Verify tables created: Go to **Table Editor** → Should see 3 tables

1.4 Test Database Connection

1. Go to **Table Editor** → **leads**
2. Click **"Insert row"**
3. Add test data:
   - full_name: "Test Lead"
   - phone: "555-555-5555"
   - message: "Test message"
4. Click **"Save"**
5. Verify row appears
6. Delete test row

✅ **Supabase setup complete**

STEP 2: PURCHASE TWILIO NUMBER (5 minutes)

2.1 Buy Phone Number

1. Go to https://console.twilio.com
2. Click **Phone Numbers** → **Buy a number**
3. Search criteria:
   - **Country:** United States
   - **Area Code:** [Client's preferred area code from form]
   - **Capabilities:**
     - ☑ Voice
     - ☑ SMS
     - ☑ MMS
4. Click **Search**
5. Select a number with good digits (avoid hard-to-remember numbers)
6. Click **Buy**
7. Confirm purchase ($2.75/month)

2.2 Save Number Details

Copy to `PROJECT_SECRETS_REFERENCE.txt`:

```
Twilio Phone Number: +1 (404) 555-0123
Twilio Number SID: PNxxxxxxxxxxxx
Purchased: [date]
Forwards to: [client's real phone number]
```

2.3 Configure Voice Webhook (Do after Vercel deployment)

Skip this for now - we'll come back after deploying to Vercel

2.4 Configure SMS Webhook (Do after Vercel deployment)

Skip this for now - we'll come back after deploying to Vercel

✅ **Twilio number purchased (configuration pending deployment)**

STEP 3: PREPARE CLIENT FILES (10 minutes)

3.1 Create Client Config File

1. In your template repo, create: `/src/config/clients/[client-slug].ts`

2. Example: `/src/config/clients/atlanta-hvac.ts`

3. Copy this template and fill in from form responses:

```
export const atlantaHVAC = {
  // Basic Info
  slug: 'atlanta-hvac',
  businessName: 'Atlanta Emergency HVAC',
  tagline: '24/7 Emergency HVAC Services',
  industry: 'hvac',

  // Contact
  email: 'service@atlantahvac.com',
  phone: '(404) 555-0100', // Their real number
  trackingPhone: '(404) 555-0123', // Twilio number
  address: '123 Main St, Atlanta, GA 30303',
  city: 'Atlanta',
  state: 'GA',
  // Branding
  logo: '/clients/atlanta-hvac/logo.png',
  primaryColor: '#234654',
  secondaryColor: '#F5A05C',

  // Hero Section
  hero: {
    headline: '24/7 Emergency HVAC Services in Atlanta',
    subheadline: 'Fast Response • Expert Technicians • Fair Pricing',
    description: 'When your AC breaks in the Georgia heat, you need help FAST. Our certified
technicians respond within 1 hour for emergency service.',
    image: '/clients/atlanta-hvac/hero.jpg'
  },

  // About Section
  about: {
    heading: 'about us',
    description: 'With over 20 years of experience, Atlanta Emergency HVAC specializes in fast,
reliable HVAC repair and installation. We serve homeowners throughout the Atlanta metro area
with professional service you can trust.',
    image: '/clients/atlanta-hvac/about.jpg'
  },

  // Services (8 services)
  services: [
    { title: 'Emergency AC Repair', icon: 'snowflake' },
    { title: 'Emergency Heating Repair', icon: 'flame' },
    { title: 'AC Installation', icon: 'wind' },
    { title: 'Heating Installation', icon: 'thermometer' },
    { title: 'Maintenance Plans', icon: 'wrench' },
    { title: 'Duct Cleaning', icon: 'wind' },
    { title: 'Thermostat Install', icon: 'gauge' },
    { title: 'Air Quality Services', icon: 'leaf' }
  ],

  // Trust Badges (5 badges)
  trustBadges: [
    '24/7 Emergency Service',
    'Licensed & Insured',
    '20+ Years Experience',
    'Same-Day Service',
    '100% Satisfaction Guaranteed'
  ],
  // Process Steps
  process: {
    heading: 'our process',
    subheading: 'Getting your HVAC fixed is easy',
    steps: [
      { icon: 'phone', title: 'Call or Text Us' },
      { icon: 'calendar', title: 'Schedule Service' },
      { icon: 'wrench', title: 'We Fix It Fast' },
      { icon: 'clipboard-check', title: 'Quality Inspection' },
      { icon: 'handshake', title: 'Pay After Satisfied' }
    ]
  },

  // Gallery Images (array of image paths)
  gallery: {
    images: [
      { url: '/clients/atlanta-hvac/gallery/1.jpg', alt: 'AC Repair' },
      { url: '/clients/atlanta-hvac/gallery/2.jpg', alt: 'Furnace Install' },
      { url: '/clients/atlanta-hvac/gallery/3.jpg', alt: 'HVAC Maintenance' },
      { url: '/clients/atlanta-hvac/gallery/4.jpg', alt: 'Duct Cleaning' },
      { url: '/clients/atlanta-hvac/gallery/5.jpg', alt: 'Thermostat Install' },
      { url: '/clients/atlanta-hvac/gallery/6.jpg', alt: 'AC Installation' },
      { url: '/clients/atlanta-hvac/gallery/7.jpg', alt: 'Heat Pump Repair' },
      { url: '/clients/atlanta-hvac/gallery/8.jpg', alt: 'Emergency Service' }
    ]
  },

  // Reviews
  reviews: {
    googleUrl: 'https://g.page/atlanta-hvac',
    embedCode: '<iframe src="..."></iframe>' // From form
  },

  // FAQs (3-5 questions)
  faqs: [
    {
      question: 'How quickly can you respond to emergencies?',
      answer: 'We respond to emergency calls within 1 hour. Our technicians are on-call 24/7,
including weekends and holidays.'
    },
    {
      question: 'Do you offer financing options?',
      answer: 'Yes, we offer flexible financing options for new system installations and major repairs. Ask us about our current promotions.'
    },
    {
      question: 'Are your technicians certified?',
      answer: 'Absolutely. All our technicians are EPA certified, licensed, and undergo continuous training to stay updated on the latest HVAC technology.'
    }
  ],

  // Service Areas
  serviceAreas: {
    mapEmbed: '<iframe src="https://maps.google.com/..."></iframe>',
    areas: [
      'Midtown Atlanta',
      'Buckhead',
      'Virginia Highland',
      'Inman Park',
      'Decatur',
      'Sandy Springs',
      'Brookhaven',
      'Dunwoody'
    ]
  },

  // Business Hours
  hours: {
    'Mon': '24/7 Emergency Service',
    'Tue': '24/7 Emergency Service',
    'Wed': '24/7 Emergency Service',
    'Thu': '24/7 Emergency Service',
    'Fri': '24/7 Emergency Service',
    'Sat': '24/7 Emergency Service',
    'Sun': '24/7 Emergency Service'
  },

  // SMS Auto-Response
  smsAutoResponse: 'Thanks for contacting Atlanta Emergency HVAC! We received your message and a technician will call you back within 30 minutes. For immediate emergency service, call us at (404) 555-0100.',

  // Supabase Connection (from Step 1)
  supabase: {
    url: 'https://xxxxx.supabase.co',
    anonKey: 'eyJhbGc...'
  },

  // Social Links (optional)
  socialLinks: [
    { platform: 'facebook', url: 'https://facebook.com/atlantahvac' },
    { platform: 'instagram', url: 'https://instagram.com/atlantahvac' }
  ]
};
```

3.2 Upload Client Assets

1. Create folder: `/public/clients/[client-slug]/`
    - Example: `/public/clients/atlanta-hvac/`

Upload files from form responses:

```
/public/clients/atlanta-hvac/
├── logo.png
├── hero.jpg
├── about.jpg
└── gallery/
    ├── 1.jpg
    ├── 2.jpg
    ├── 3.jpg
    ├── 4.jpg
    ├── 5.jpg
    ├── 6.jpg
    ├── 7.jpg
    └── 8.jpg
```

2.
3. Optimize images (if needed):
    - Logo: <100KB, transparent background
    - Hero: <500KB, 1920x1080
    - Gallery: <300KB each, 1200x800

3.3 Create Environment Variables File

Create `.env.local` (don't commit this!):

```
# Supabase (from Step 1)
NEXT_PUBLIC_SUPABASE_URL=https://xxxxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJhbGc...
SUPABASE_SERVICE_ROLE_KEY=eyJhbGc...

# Twilio (from Step 2)
TWILIO_ACCOUNT_SID=ACxxxxxxxxxxxx
TWILIO_AUTH_TOKEN=your_auth_token
TWILIO_PHONE_NUMBER=+14045550123

# Client Config
CLIENT_SLUG=atlanta-hvac

# App URL (update after deployment)
NEXT_PUBLIC_APP_URL=https://atlantahvac.com
```

✅ **Client files prepared**

STEP 4: DEPLOY TO VERCEL (5 minutes)

4.1 Commit Changes to Git

```
git add .
git commit -m "Add atlanta-hvac client"
git push origin main
```

4.2 Create New Vercel Project

Option A: Deploy as separate project (RECOMMENDED for isolation)

1. Go to https://vercel.com/dashboard
2. Click **"Add New..."** → **"Project"**
3. **Import Git Repository:**
    - Select your template repo
    - Click **"Import"**
4. **Configure Project:**
    - **Project Name:** `atlanta-hvac-website`
    - **Framework Preset:** Next.js (auto-detected)
    - **Root Directory:** `./` (leave as is)
5. **Environment Variables:**

  - Click **"Add"** for each variable from `.env.local`:
    - `NEXT_PUBLIC_SUPABASE_URL`
    - `NEXT_PUBLIC_SUPABASE_ANON_KEY`
    - `SUPABASE_SERVICE_ROLE_KEY`
    - `TWILIO_ACCOUNT_SID`
    - `TWILIO_AUTH_TOKEN`
    - `TWILIO_PHONE_NUMBER`
    - `CLIENT_SLUG`
    - `NEXT_PUBLIC_APP_URL` (use temporary vercel URL for now)

6. Click **"Deploy"**
7. Wait 2-3 minutes for build

Option B: Deploy as subdomain (cheaper but less isolated)

- Use single Vercel project
- Deploy to `atlanta-hvac.emergencyhvacleads.com`
- NOT RECOMMENDED for true client isolation

4.3 Get Vercel URL

After deployment:

1. Copy the Vercel URL: `https://atlanta-hvac-website.vercel.app`
2. Test it: Open in browser, verify site loads

4.4 Configure Custom Domain (if client has one)

1. In Vercel project, go to **Settings** → **Domains**
2. Click **"Add"**
3. Enter client's domain: `atlantahvac.com`
4. Vercel will show DNS records to configure

✅ **Deployed to Vercel**

STEP 5: CONFIGURE DNS (5 minutes)

5.1 Get DNS Records from Vercel

From Step 4.4, copy these records:

For Root Domain (atlantahvac.com):

- Type: `A`
- Name: `@`
- Value: `76.76.19.19`

For WWW Subdomain (www.atlantahvac.com):

- Type: `CNAME`
- Name: `www`
- Value: `cname.vercel-dns.com`

5.2 Update DNS at Domain Registrar

1. Log into client's domain registrar (Porkbun, GoDaddy, Namecheap, etc.)
2. Go to DNS settings
3. Add/update records from Step 5.1
4. Save changes

5.3 Wait for DNS Propagation

- Takes 5 minutes to 48 hours (usually 15-30 minutes)
- Check status: https://dnschecker.org
- Enter domain: `atlantahvac.com`
- Wait for green checkmarks globally

5.4 Verify SSL Certificate

1. Once DNS propagates, Vercel auto-issues SSL
2. Check: `https://atlantahvac.com` should work with padlock icon
3. If SSL pending, give it 10 more minutes

✅ **Domain configured and live**

STEP 6: CONFIGURE TWILIO WEBHOOKS (5 minutes)

Now that site is live, configure Twilio to send calls/SMS to your API.

6.1 Get Webhook URLs

Your site is now live at: `https://atlantahvac.com`

Webhook URLs:

- Voice: `https://atlantahvac.com/api/twilio/voice`
- SMS: `https://atlantahvac.com/api/twilio/sms`
- Status: `https://atlantahvac.com/api/twilio/status`

6.2 Configure Voice Webhook

1. Go to https://console.twilio.com
2. Click **Phone Numbers** → **Manage** → **Active Numbers**
3. Click the number you purchased in Step 2
4. Scroll to **Voice Configuration:**
   - **A CALL COMES IN:** Webhook
   - **URL:** `https://atlantahvac.com/api/twilio/voice`
   - **HTTP:** POST
   - **Primary Handler Fails:** Leave blank (or add fallback URL)
5. Scroll to **Call Status Changes:**
   - **URL:** `https://atlantahvac.com/api/twilio/status`
   - **HTTP:** POST
6. Click **"Save configuration"**

6.3 Configure SMS Webhook

1. Same page, scroll to **Messaging Configuration:**
   - **A MESSAGE COMES IN:** Webhook
   - **URL:** `https://atlantahvac.com/api/twilio/sms`
   - **HTTP:** POST
2. Click **"Save configuration"**

6.4 Test Twilio Integration

Test Call:

1. Call the Twilio tracking number from your phone
2. Should forward to client's real number
3. Check Supabase → `call_logs` table → verify row created

Test SMS:

1. Text the Twilio tracking number from your phone
2. Should receive auto-response SMS
3. Check Supabase → `sms_logs` table → verify row created

✅ **Twilio configured and working**

STEP 7: FINAL TESTING (5 minutes)

7.1 Test Form Submission

1. Go to `https://atlantahvac.com`
2. Fill out the contact form
3. Submit
4. Check Supabase → `leads` table → verify row created
5. Verify SMS sent to phone number entered

7.2 Test All Pages

- [ ] Homepage loads
- [ ] All images display
- [ ] All sections visible
- [ ] Mobile responsive (test on phone)
- [ ] Form submits successfully
- [ ] No console errors (check browser dev tools)

7.3 Test Phone/SMS

- [ ] Call tracking number → forwards correctly
- [ ] Miss a call → auto-SMS sent
- [ ] Text tracking number → auto-response received
- [ ] All logs appear in Supabase

7.4 Performance Check

1. Go to https://pagespeed.web.dev
2. Enter: `https://atlantahvac.com`
3. Run test
4. Target scores:
   - Performance: 80+
   - Accessibility: 90+
   - Best Practices: 90+
   - SEO: 90+

✅ **All tests passing**

STEP 8: DELIVER TO CLIENT (5 minutes)

8.1 Send Client Email

Subject: Your Website is Live! 🎉

Hi [Client Name],

Great news! Your website is now live at:
https://atlantahvac.com

Your tracking phone number:
(404) 555-0123

This number forwards to your main line: (404) 555-0100

What happens now:
✓ When customers call your tracking number, it forwards to you
✓ If you miss a call, they automatically get a text message
✓ When someone fills out your contact form, their info is saved and you get notified
✓ All leads are tracked in your dashboard (login info below)

Important Numbers:
- Tracking Number: (404) 555-0123 (use this on ALL marketing)
- Your Real Number: (404) 555-0100 (calls forward here)

Dashboard Login (coming soon):
- URL: https://atlantahvac.com/dashboard
- Email: [their email]
- Password: [generate temporary password]

Monthly Billing:
- Starts: [date]
- Amount: $297/month
- Payment method: [from form]

Need any changes? Just reply to this email.

Welcome aboard!

Jeff Chery
Chery Solutions LLC
jeff@cherysolutions.com
[Your Phone]

8.2 Update Internal Tracking

Add to your client spreadsheet:

- Client name
- Domain
- Tracking phone number
- Supabase project URL
- Launch date
- Monthly billing date
- Status: Active

8.3 Schedule 48-Hour Follow-Up

Set reminder to check in 48 hours:

- "How's the site working?"
- "Getting any calls/leads?"
- "Need any changes?"

✅ **Client delivered and happy**

TROUBLESHOOTING

Issue: Vercel build fails

Solution:

- Check build logs in Vercel
- Verify all environment variables set
- Check for typos in client config file
- Test build locally: `npm run build`

Issue: DNS not propagating

Solution:

- Wait longer (up to 48 hours max)
- Clear browser cache
- Test on different device/network
- Verify DNS records are correct (dnschecker.org)

Issue: SSL certificate pending

Solution:

- Wait 10-15 minutes after DNS propagates
- Vercel auto-issues SSL, just be patient
- If >1 hour, contact Vercel support

Issue: Twilio calls not forwarding

Solution:

- Verify webhook URL is correct
- Check Twilio logs: Console → Monitor → Logs
- Test webhook endpoint manually
- Verify client's real phone number accepts calls

Issue: Form not submitting

Solution:

- Check browser console for errors
- Verify Supabase credentials in Vercel env vars
- Test Supabase connection: Go to API route directly
- Check Supabase table permissions

Issue: Images not loading

Solution:

- Verify images in `/public/clients/[slug]/` folder
- Check file names match config file exactly
- Optimize large images (<500KB)
- Clear cache and hard refresh

OPTIMIZATION (Optional)

After First 5 Clients

Consider automating:

1. **Bash script to create Supabase tables**
2. **Script to generate client config from form responses**
3. **Automated Vercel deployment via API**

4. **DNS configuration via Cloudflare API**

After 20 Clients

Consider:

1. **Central admin dashboard** to manage all clients
2. **Automated billing** via Stripe subscriptions
3. **Client dashboard** for viewing their leads/calls
4. **Hire VA** to handle client onboarding

CHECKLIST SUMMARY

For each new client:

Supabase (5 min):

- [ ] Create project
- [ ] Run SQL to create tables
- [ ] Save credentials

Twilio (5 min):

- [ ] Buy phone number
- [ ] Save number details
- [ ] Configure webhooks (after deployment)

Files (10 min):

- [ ] Create config file
- [ ] Upload images
- [ ] Create .env.local

Deploy (5 min):

- [ ] Commit to Git
- [ ] Deploy to Vercel
- [ ] Add environment variables

DNS (5 min):

- [ ] Configure domain registrar
- [ ] Wait for propagation
- [ ] Verify SSL

Twilio Setup (5 min):

- [ ] Add voice webhook
- [ ] Add SMS webhook
- [ ] Test calls and SMS

Test (5 min):

- [ ] Form submission
- [ ] Call forwarding
- [ ] SMS auto-response
- [ ] Mobile responsive

Deliver (5 min):

- [ ] Email client
- [ ] Update tracking sheet
- [ ] Schedule follow-up

Total: 30 minutes per client (once you're fast)

END OF SETUP SCRIPT