LeadGenX Dashboard - Deployment Guide

Prerequisites

1. LeadGenX Backend API deployed and accessible
2. Domain configured (e.g., `app.leadgenx.app`)
3. Node.js 18+ on deployment environment

Environment Configuration

Production Environment Variables

Create `.env.production` or configure in deployment platform:

```
# API Configuration
NEXT_PUBLIC_API_BASE_URL=https://leadgenx.app

# Auth Secret (generate with: openssl rand -base64 32)
AUTH_SECRET=<your-production-secret>
```

Important: Change `AUTH_SECRET` in production!

Deployment Steps

Step 1: Update API URL

Update `.env.local` or platform environment variables:

```
NEXT_PUBLIC_API_BASE_URL=https://leadgenx.app
```

Or if backend is on subdomain:

```
NEXT_PUBLIC_API_BASE_URL=https://api.leadgenx.app
```

Step 2: Build for Production

```
cd /home/ubuntu/leadgenx-dashboard
npm install
npm run build
```

Verify build succeeds:

```
✓ Compiled successfully
✓ Running TypeScript
✓ Generating static pages
✓ Finalizing page optimization

Route (app)                      Size
┌ ○ /                            XXX kB
├ ○ /campaigns                   XXX kB
├ ƒ /campaigns/[id]              XXX kB
├ ○ /clients                     XXX kB
├ ○ /export                      XXX kB
└ ○ /leads                       XXX kB
```

Step 3: Test Locally

```
npm start
```

Open `http://localhost:3000` and verify:

1. ✅ Login page loads
2. ✅ Can login with API key
3. ✅ Clients page loads
4. ✅ Can create a campaign
5. ✅ API calls work

Step 4: Deploy to Abacus AI

Option A: Via UI

1. Go to Abacus AI Dashboard
2. Navigate to deployments
3. Select “New Deployment”
4. Choose “Next.js Application”
5. Upload project or connect Git
6. Set environment variables
7. Set domain: `app.leadgenx.app`
8. Deploy

Option B: Via CLI

```
# Install Abacus CLI (if not installed)
npm install -g @abacusai/cli

# Login
abacus login

# Deploy
abacus deploy --project leadgenx-dashboard --domain app.leadgenx.app
```

Step 5: Configure DNS

If using custom domain:

1. Go to domain registrar (where you bought `leadgenx.app`)

2. Add DNS record:
   - Type: `CNAME`
   - Name: `app`
   - Value: `<provided-by-abacus>`
   - TTL: `300` (5 minutes)

3. Wait for DNS propagation (5-30 minutes)

4. Verify:
   `bash`
   `curl -I https://app.leadgenx.app`

Step 6: Verify Deployment

1. Visit `https://app.leadgenx.app`
2. Login with test API key
3. Test all features:
   - ✅ Clients CRUD
   - ✅ Campaign creation
   - ✅ Campaign run
   - ✅ Leads display
   - ✅ Export functionality

Domain Architecture

Recommended Setup

```
leadgenx.app              → Marketing/Landing page (future)
api.leadgenx.app          → Backend API (Swagger docs)
app.leadgenx.app          → Dashboard (this project)
```

Current Setup

```
leadgenx.app              → Backend API
app.leadgenx.app          → Dashboard (to be deployed)
```

After backend moves to `api.leadgenx.app`, update dashboard env:

```
NEXT_PUBLIC_API_BASE_URL=https://api.leadgenx.app
```

Redeploy dashboard with new env var.

Deployment Checklist

Pre-Deployment

- [ ] Backend API is deployed and accessible
- [ ] Environment variables configured
- [ ] Production build succeeds locally
- [ ] Local production test passes

- [ ] API key authentication works
- [ ] All pages load without errors

Deployment

- [ ] Project deployed to platform
- [ ] Domain configured correctly
- [ ] DNS records updated (if custom domain)
- [ ] HTTPS certificate active
- [ ] Environment variables set on platform

Post-Deployment

- [ ] Dashboard accessible at production URL
- [ ] Login works
- [ ] API calls succeed
- [ ] All features tested
- [ ] No console errors
- [ ] Performance acceptable (Lighthouse score > 80)

Monitoring

Health Checks

```
# Check dashboard is up
curl -I https://app.leadgenx.app

# Check API connectivity
curl https://app.leadgenx.app/api/health
```

Logs

Abacus AI Platform:
- View logs in dashboard
- Filter by severity
- Search for errors

Browser Console:
- Check for JavaScript errors
- Monitor network requests
- Verify API responses

Performance

```
# Run Lighthouse audit
npx lighthouse https://app.leadgenx.app --view
```

Target scores:
- Performance: 80+
- Accessibility: 90+
- Best Practices: 90+
- SEO: 80+

Rollback

If deployment fails:

1. **Identify issue** in logs
2. **Revert to previous version** via platform UI
3. **Fix locally** and test
4. **Redeploy** when ready

Continuous Deployment

Git Integration

Connect repository for auto-deploy:

1. Link GitHub/GitLab repo
2. Set production branch (e.g., `main`)
3. Configure build command: `npm run build`
4. Set start command: `npm start`
5. Enable auto-deploy on push

CI/CD Pipeline

```yaml
# Example GitHub Actions workflow
name: Deploy Dashboard

on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm install
      - run: npm run build
      - run: npm test
      - name: Deploy to Abacus
        run: npx @abacusai/cli deploy
        env:
          ABACUS_API_KEY: ${{ secrets.ABACUS_API_KEY }}
```

Troubleshooting

Build Fails

```bash
# Clear cache
rm -rf .next node_modules
npm install
npm run build
```

API Calls Fail

1. Check `NEXT_PUBLIC_API_BASE_URL` is correct
2. Verify backend is accessible:
   `bash`
   `  curl https://leadgenx.app/health`
3. Check CORS headers on backend
4. Verify API key is valid

DNS Issues

```
# Check DNS propagation
dig app.leadgenx.app
nslookup app.leadgenx.app

# Use different DNS server
dig @8.8.8.8 app.leadgenx.app
```

SSL Certificate Issues

- Wait for automatic certificate provisioning (5-10 minutes)
- Check certificate status in platform dashboard
- Verify DNS is pointing to correct target

Security Considerations

Environment Variables

- ✅ Never commit `.env.local` to git
- ✅ Use different `AUTH_SECRET` in production
- ✅ Rotate API keys regularly
- ✅ Use HTTPS only in production

API Security

- Backend must validate API keys
- Implement rate limiting
- Use CORS correctly
- Sanitize user inputs

Client-Side Security

- API keys stored in localStorage (acceptable for this use case)
- No sensitive data in URLs
- XSS protection via React
- CSRF not applicable (API key auth)

Support

If deployment issues persist:

1. Check Abacus AI Documentation (https://docs.abacus.ai)
2. Review backend logs at `https://leadgenx.app/docs`

3. Contact: support@leadgenx.app

Next Steps

After successful deployment:

1. **Share access** with team members (provide API keys)
2. **Monitor usage** and performance
3. **Collect feedback** from users
4. **Plan Phase 12** (Advanced Features)
5. **Integrate with CRM** (Zapier/HubSpot)