

Narrative News - Complete Deployment Guide

This guide will take you from development to a fully operational production website.

Pre-Deployment Checklist

Required Accounts & Services

- ☐ **Domain Name** - Purchase your domain (narrativenews.org)
- ☐ **Hosting Platform** - Vercel (recommended) or Railway
- ☐ **Database** - Supabase (recommended) or Railway PostgreSQL
- ☐ **OpenAI Account** - For AI analysis generation
- ☐ **Email Service** - Gmail App Password or SendGrid
- ☐ **Google AdSense** - For revenue generation
- ☐ **Google Analytics** - For tracking (optional)

Step-by-Step Deployment

Step 1: Database Setup (Supabase - Recommended)

1. Create Supabase Project

```
bash

# Go to https://supabase.com/dashboard
# Click "New Project"
# Choose organization and enter project details
# Wait for database to be ready (2-3 minutes)
```

2. Get Database Connection String

```
bash

# In Supabase Dashboard > Settings > Database
# Copy the connection string
# Format: postgresql://postgres:[PASSWORD]@db.[REF].supabase.co:5432/postgres
```

3. Run Database Migrations

```
bash
```

```
# Set your DATABASE_URL
export DATABASE_URL="your-supabase-connection-string"

# Generate Prisma client
npx prisma generate

# Push schema to database
npx prisma db push

# Seed initial data
npx prisma db seed
```

Step 2: OpenAI API Setup

1. Get OpenAI API Key

```
bash

# Go to https://platform.openai.com/api-keys
# Click "Create new secret key"
# Copy the key (starts with sk-)
# Add billing method if not already done
```

2. Test API Connection

```
bash

# Test your key works
curl -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"model": "gpt-3.5-turbo", "messages": [{"role": "user", "content": "test"}], "max_tokens": 5}' \
  https://api.openai.com/v1/chat/completions
```

Step 3: Email Service Setup (Gmail)

1. Enable 2-Factor Authentication

```
bash

# Go to myaccount.google.com/security
# Turn on 2-Step Verification
```

2. Generate App Password

```
bash
```

```
# Go to myaccount.google.com/apppasswords
```

```
# Select Mail and your device
```

```
# Copy the 16-character app password
```

3. Configure Email Settings

```
env
```

```
EMAIL_SERVER_HOST=smtp.gmail.com
```

```
EMAIL_SERVER_PORT=587
```

```
EMAIL_SERVER_USER=your-email@gmail.com
```

```
EMAIL_SERVER_PASSWORD=your-16-char-app-password
```

```
EMAIL_FROM="Narrative News <noreply@your-domain.com>"
```

Step 4: AdSense Setup

1. Apply for AdSense

```
bash
```

```
# Go to https://www.google.com/adsense/
```

```
# Apply with your domain
```

```
# Wait for approval (can take days/weeks)
```

2. Get Publisher ID

```
bash
```

```
# Once approved, go to AdSense dashboard
```

```
# Account > Settings > Account information
```

```
# Copy Publisher ID (ca-pub-xxxxxxxxxxxxxxxxxx)
```

3. Create Ad Units

```
bash
```

```
# AdSense > Ads > By ad unit
```

```
# Create display ads for:
```

```
# - Banner (728x90 or responsive)
```

```
# - Sidebar (300x250 or responsive)
```

```
# - In-article (responsive)
```

```
# - Footer (728x90 or responsive)
```

```
# Copy each ad unit ID
```

Step 5: Vercel Deployment (Recommended)

1. Install Vercel CLI

```
bash  
  
npm install -g vercel
```

2. Connect GitHub Repository

```
bash  
  
# Push your code to GitHub  
git add .  
git commit -m "Ready for deployment"  
git push origin main
```

3. Deploy to Vercel

```
bash  
  
# Login to Vercel  
vercel login  
  
# Deploy project  
vercel --prod  
  
# Or deploy from GitHub:  
# Go to https://vercel.com/new  
# Import your GitHub repository  
# Configure environment variables  
# Deploy
```

4. Configure Environment Variables

```
bash  
  
# In Vercel Dashboard > Settings > Environment Variables  
# Add all variables from .env.production.template  
# Make sure to set NODE_ENV=production
```

5. Configure Custom Domain

```
bash
```

```
# In Vercel Dashboard > Settings > Domains
# Add your custom domain
# Update DNS records as instructed
# Enable SSL (automatic)
```

Step 6: Alternative - Railway Deployment

1. Install Railway CLI

```
bash

npm install -g @railway/cli
```

2. Deploy to Railway

```
bash

# Login
railway login

# Initialize project
railway init

# Add PostgreSQL database
railway add postgresql

# Deploy
railway up

# Set environment variables
railway variables set NODE_ENV=production
railway variables set DATABASE_URL=$DATABASE_URL
# ... add all other environment variables
```

Production Configuration

Environment Variables Setup

Create `.env.local` with production values:

```
env
```

```
# Copy from .env.production.template
# Fill in all actual values
DATABASE_URL="your-actual-database-url"
OPENAI_API_KEY="your-actual-openai-key"
EMAIL_SERVER_USER="your-actual-email"
# ... etc
```

SSL Certificate Setup

```
bash

# Vercel handles this automatically
# For custom hosting, use Let's Encrypt:
sudo certbot --nginx -d narrativenews.org -d www.narrativenews.org
```

CDN Configuration

```
bash

# Vercel provides global CDN automatically
# For other platforms, configure CloudFlare:
# 1. Add site to CloudFlare
# 2. Update nameservers
# 3. Enable proxy for main domain
# 4. Configure caching rules
```

Post-Deployment Setup

1. Test Core Functionality

```
bash

# Test homepage
curl -I https://your-domain.com

# Test API endpoints
curl https://your-domain.com/api/articles

# Test admin dashboard
# Visit https://your-domain.com/admin
```

2. Configure Automation

```
bash

# Access admin dashboard
# Navigate to Automation tab
# Start automation system
# Verify RSS feeds are working
# Test email configuration
```

3. Set Up Monitoring

```
bash

# Add Sentry for error tracking (optional)
npm install @sentry/nextjs
# Configure in next.config.js

# Set up uptime monitoring
# Use services like UptimeRobot or Pingdom
# Monitor: Homepage, API health, admin dashboard
```

4. Content Management

```
bash

# Seed initial RSS feeds
# Categories to cover:
# - Politics: CNN, Fox News, Reuters
# - Business: WSJ, Business Insider
# - Technology: TechCrunch, The Verge
# - General: BBC, AP News

# Test article generation
# Verify AI analysis quality
# Check email newsletter formatting
```

Performance Optimization

Database Optimization

```
sql
```

```
-- Add indexes for common queries
```

```
CREATE INDEX IF NOT EXISTS idx_articles_published_at ON articles(published_at DESC);
```

```
CREATE INDEX IF NOT EXISTS idx_articles_category ON articles(category);
```

```
CREATE INDEX IF NOT EXISTS idx_subscribers_active ON subscribers(is_active);
```

```
CREATE INDEX IF NOT EXISTS idx_rss_feeds_active ON rss_feeds(is_active);
```

Caching Strategy

```
javascript
```

```
// Add Redis caching (optional)
```

```
// Cache article lists for 5 minutes
```

```
// Cache individual articles for 1 hour
```

```
// Cache RSS feed results for 15 minutes
```

Image Optimization

```
javascript
```

```
// Configure next.config.js for optimal images
```

```
// Use WebP format when supported
```

```
// Lazy load images below the fold
```

```
// Optimize AdSense ad images
```



Security Hardening

API Security

```
javascript
```

```
// Rate limiting is already configured
```

```
// Additional security measures:
```

```
// - API key rotation schedule
```

```
// - Input validation on all endpoints
```

```
// - CORS policy refinement
```

```
// - Request size limits
```

Database Security

```
bash
```



```
# Configure connection pooling
# Set up read replicas if high traffic
# Enable connection encryption
# Regular security updates
```

Content Security

```
javascript

// CSP headers in next.config.js
// XSS protection enabled
// SQL injection prevention via Prisma
// Input sanitization for user content
```

SEO & Marketing Setup

Search Console

```
bash

# Add site to Google Search Console
# Submit sitemap: https://your-domain.com/sitemap.xml
# Monitor indexing status
# Set up email notifications
```

Social Media Integration

```
html

<!-- Open Graph tags already included -->
<!-- Twitter cards configured -->
<!-- Schema markup for articles -->
```

Analytics Setup

```
javascript
```

```
// Google Analytics 4 configured
// Track key events:
// - Article views
// - Newsletter signups
// - Source link clicks
// - Time on page
```

Troubleshooting Guide

Common Deployment Issues

Database Connection Errors

```
bash

# Check connection string format
# Verify database is accessible
# Check firewall settings
# Test with: npx prisma db pull
```

OpenAI API Errors

```
bash

# Verify API key is correct
# Check billing setup
# Test rate limits
# Monitor usage quotas
```

Email Delivery Issues

```
bash

# Verify SMTP credentials
# Check spam folder
# Test with: POST /api/email/test
# Monitor email reputation
```

Automation Not Running

```
bash
```

```
# Check logs in admin dashboard
# Verify cron jobs are scheduled
# Test RSS feed accessibility
# Check rate limiting
```

Performance Issues

Slow Page Loading

```
bash

# Check Lighthouse score
# Optimize images
# Enable compression
# Review database queries
```

High Memory Usage

```
bash

# Monitor heap usage
# Optimize database connections
# Review RSS scraping frequency
# Check for memory leaks
```

Go-Live Checklist

Pre-Launch

- ☐ Domain configured and SSL working
- ☐ All environment variables set
- ☐ Database migrations completed
- ☐ RSS feeds configured and tested
- ☐ Email service working
- ☐ AdSense approved and ads displaying
- ☐ Admin dashboard accessible
- ☐ Automation system running
- ☐ Performance testing completed
- ☐ Security testing completed

Launch Day

- ☐ Final database backup
- ☐ DNS changes propagated
- ☐ Monitoring alerts configured
- ☐ Social media accounts ready
- ☐ Launch announcement prepared
- ☐ Support email configured
- ☐ Analytics tracking verified

Post-Launch (Week 1)

- ☐ Monitor error rates
- ☐ Check email deliverability
- ☐ Verify ad revenue tracking
- ☐ Review automation performance
- ☐ Monitor user feedback
- ☐ Check search engine indexing
- ☐ Analyze initial traffic patterns

Support & Maintenance

Regular Maintenance Tasks

Daily

- Check automation status
- Monitor error logs
- Verify newsletter delivery

Weekly

- Review analytics
- Check AdSense earnings
- Update RSS feeds if needed
- Database performance review

Monthly

- Security updates
- Performance optimization

- Content quality review
- Backup verification

Getting Help

Technical Issues

- Check admin dashboard health status
- Review application logs
- Test individual components
- Monitor third-party service status

Business Issues

- AdSense policy compliance
- Email deliverability reputation
- Content quality guidelines
- User feedback management

Congratulations!

Your Narrative News platform is now live! You have built a complete, automated news analysis platform that:

- ✓ **Automatically scrapes news** from multiple sources
- ✓ **Generates AI-powered analysis** comparing different perspectives
- ✓ **Sends daily newsletters** to subscribers
- ✓ **Generates revenue** through strategic ad placement
- ✓ **Provides admin control** over all aspects of the platform
- ✓ **Scales automatically** with your growing audience

Your platform is now ready to provide unbiased news analysis to readers while generating revenue through your hands-off automation system.

Next Steps:

1. Promote your platform to build your initial audience
2. Monitor performance and optimize based on user behavior
3. Consider adding premium features as you grow

4. Expand to additional news categories and sources

Welcome to the world of automated journalism! 🚀