



Narrative News

AI-powered news analysis platform that reveals how different outlets frame the same stories

Narrative News automatically scrapes articles from left and right-leaning news sources, then uses AI to generate unbiased analysis comparing how each side presents the same events. Get the full picture of current events without the bias.

[Show Image](#)

[Show Image](#)

✨ Features



Automated Content Generation

- **Smart News Scraping:** Automatically monitors RSS feeds from major news outlets
- **AI-Powered Analysis:** Uses OpenAI GPT-4 to compare different perspectives on the same story
- **Duplicate Detection:** Prevents republishing the same content
- **Content Matching:** Intelligently pairs articles covering the same events



Revenue Generation

- **Newsletter Automation:** Daily email campaigns with beautiful HTML templates
- **Google AdSense Integration:** Strategic ad placement for maximum revenue
- **Subscriber Management:** Automated email collection and management
- **Premium Ready:** Built-in infrastructure for paid subscriptions



Admin Dashboard

- **System Monitoring:** Real-time health checks and performance metrics
- **Content Management:** Bulk operations on articles and sources
- **Analytics:** Traffic, engagement, and revenue tracking

- **Automation Control:** Start, stop, and monitor automated processes

Analytics & Insights

- **Traffic Analysis:** Page views, user engagement, source preferences
- **Revenue Tracking:** Ad performance and subscriber growth
- **Content Performance:** Most popular articles and categories
- **Source Balance:** Monitor left vs right source coverage

Quick Start

Prerequisites

- Node.js 18+
- PostgreSQL database (or Supabase)
- OpenAI API key
- Email service (Gmail/SendGrid)
- Google AdSense account (for revenue)

One-Click Deploy

Vercel (Recommended)

```
bash
```

Fork this repository, then:

<https://vercel.com/new/clone?repository-url=https://github.com/yourusername/narrative-news>

Railway

```
bash
```

Deploy to Railway with database included:

<https://railway.app/template/narrative-news>

Local Development

```
bash
```

```
# Clone the repository
git clone https://github.com/yourusername/narrative-news.git
cd narrative-news

# Install dependencies
npm install

# Set up environment variables
cp .env.example .env.local
# Edit .env.local with your configuration

# Set up database
npx prisma generate
npx prisma db push
npx prisma db seed

# Start development server
npm run dev
```

Visit <http://localhost:3000> to see your site.



Environment Variables

Create `.env.local` with these required variables:

```
bash
```

Database

DATABASE_URL="postgresql://user:password@host:port/database"

AI Service

OPENAI_API_KEY="sk-your-openai-key"

Email Service

EMAIL_SERVER_HOST="smtp.gmail.com"

EMAIL_SERVER_USER="your-email@gmail.com"

EMAIL_SERVER_PASSWORD="your-app-password"

EMAIL_FROM="noreply@your-domain.com"

Site Configuration

SITE_URL="https://your-domain.com"

NEXTAUTH_SECRET="your-secret-key"

Revenue (Optional)

NEXT_PUBLIC_ADSENSE_CLIENT_ID="ca-pub-your-adsense-id"

See [.env.production.template](#) for all options.

Architecture

Tech Stack

- **Framework:** Next.js 14 with TypeScript
- **Database:** PostgreSQL with Prisma ORM
- **AI:** OpenAI GPT-4 for content analysis
- **Email:** Nodemailer with SMTP
- **Styling:** Tailwind CSS
- **Charts:** Recharts for analytics
- **Automation:** Node-cron for scheduling

Project Structure

```
src/
├── app/           # Next.js app directory
│   ├── api/      # API routes
│   ├── admin/    # Admin dashboard
│   └── article/   # Article pages
└── components/   # React components
```

```
| ├── admin/      # Admin-specific components
| ├── ui/        # Reusable UI components
| ├── lib/       # Core business logic
| ├── ai.ts      # AI analysis service
| ├── automation.ts # Automation scheduler
| ├── db.ts      # Database service
| ├── email.ts   # Email service
| ├── scraper.ts # News scraping
| ├── types/     # TypeScript definitions
| └── prisma/    # Database schema & migrations
```

Usage

Admin Dashboard

Access the admin panel at `/admin` to:

- Monitor system health and performance
- Control automation (start/stop/trigger)
- Manage RSS news sources
- View analytics and subscriber data
- Send newsletters manually

API Endpoints

- `GET /api/articles` - List articles with filtering
- `GET /api/articles/[slug]` - Get single article
- `POST /api/newsletter` - Subscribe to newsletter
- `GET /api/analytics` - Get site analytics
- `POST /api/ai-analysis` - Generate AI analysis

Automation

The system automatically:

- Scrapes RSS feeds every 6 hours
- Matches articles across different outlets
- Generates AI analysis comparing perspectives
- Sends daily newsletters at 8 AM EST

Configuration

RSS Sources

Configure news sources in the admin dashboard or directly in the database:

```
javascript

// Left-leaning sources
{ outlet: "CNN", url: "https://rss.cnn.com/rss/edition.rss", bias: "LEFT" }
{ outlet: "Guardian", url: "https://www.theguardian.com/us/rss", bias: "LEFT" }

// Right-leaning sources
{ outlet: "Fox News", url: "https://feeds.foxnews.com/foxnews/politics", bias: "RIGHT" }
{ outlet: "WSJ", url: "https://www.wsj.com/xml/rss/3_7085.xml", bias: "RIGHT" }
```

Email Templates

Customize newsletter templates in `src/lib/email.ts`:

- HTML templates with inline CSS
- Personalization variables
- Unsubscribe links
- Brand customization

Ad Placement

Configure AdSense ads in `src/components/AdSense.tsx`:

- Banner ads (top of pages)
- Sidebar ads (desktop)
- In-article ads (between content)
- Footer ads (bottom of pages)

Analytics

Built-in Analytics

- Article performance tracking
- Subscriber growth metrics
- Revenue monitoring
- Source click tracking

- User engagement analytics

Google Analytics Integration

```
javascript
```

```
// Automatic event tracking for:  
// - Page views  
// - Article reads  
// - Newsletter signups  
// - Source link clicks  
// - Ad interactions
```

Deployment

Vercel (Recommended)

1. Push code to GitHub
2. Connect repository in Vercel dashboard
3. Configure environment variables
4. Deploy automatically on push

Railway

1. Install Railway CLI: `npm install -g @railway/cli`
2. Login: `railway login`
3. Deploy: `railway up`
4. Add PostgreSQL: `railway add postgresql`

Docker

```
bash  
  
# Build image  
docker build -t narrative-news .  
  
# Run container  
docker run -p 3000:3000 --env-file .env.local narrative-news
```

See [DEPLOYMENT.md](#) for complete deployment guide.

Security

- **Rate Limiting:** API endpoints protected against abuse
- **Input Validation:** All user inputs validated with Zod
- **SQL Injection Prevention:** Prisma ORM provides protection
- **XSS Protection:** Next.js built-in XSS prevention
- **CSRF Protection:** Built-in CSRF tokens
- **Security Headers:** Comprehensive security headers configured

Scaling

Performance Optimizations

- **Database Indexing:** Optimized queries for common operations
- **Caching:** API response caching with appropriate TTLs
- **Image Optimization:** Next.js Image component with WebP
- **Bundle Optimization:** Code splitting and tree shaking

Horizontal Scaling

- **Stateless Architecture:** Easy to scale across multiple instances
- **Database Separation:** Read replicas for analytics queries
- **CDN Integration:** Static assets served from global CDN
- **Queue System:** Background job processing (Redis optional)

Contributing

We welcome contributions! Please see our [Contributing Guide](#) for details.

Development Setup

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Add tests if applicable
5. Submit a pull request

Code Quality

- **TypeScript:** Strict type checking enabled
- **ESLint:** Code linting with Next.js recommended rules

- **Prettier:** Automatic code formatting
- **Tests:** Jest testing framework configured

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.


Acknowledgments

- **OpenAI** for GPT-4 API enabling intelligent analysis
- **Vercel** for excellent Next.js hosting platform
- **Tailwind CSS** for rapid UI development
- **Prisma** for type-safe database access
- **Next.js** team for the amazing framework

Support

- **Documentation:** [Full deployment guide](#)
- **Issues:** [GitHub Issues](#)
- **Discussions:** [GitHub Discussions](#)
- **Email:** support@narrativenews.org

Built with  for unbiased journalism and automated revenue generation

 [Star this project](#) if you found it helpful!