

Narrative News - Project Structure

Module 1: Framework Foundation

Directory Structure

```
narrative-news/
├── src/
│   ├── app/                # Next.js App Router
│   │   ├── page.tsx        # Homepage
│   │   ├── layout.tsx      # Root layout
│   │   ├── globals.css     # Global styles
│   │   └── api/            # API routes
│   │       ├── articles/   # Article endpoints
│   │       ├── automation/ # Automation triggers
│   │       └── newsletter/ # Email signup
│   ├── components/        # Reusable UI components
│   │   ├── ui/            # Basic UI elements
│   │   ├── ArticleCard.tsx # Article display
│   │   ├── NewsAnalysis.tsx # AI analysis display
│   │   └── Newsletter.tsx  # Email signup form
│   ├── lib/              # Utility functions
│   │   ├── db.ts         # Database connection
│   │   ├── ai.ts         # AI integration
│   │   ├── scraper.ts    # News scraping
│   │   └── email.ts      # Email service
│   ├── types/            # TypeScript definitions
│   │   └── index.ts      # Global types
│   └── hooks/            # Custom React hooks
├── prisma/
│   ├── schema.prisma     # Database schema
│   └── migrations/       # Database migrations
├── public/              # Static assets
├── .env.local           # Environment variables
├── .env.example         # Env template
├── next.config.js       # Next.js configuration
├── tailwind.config.js   # Tailwind CSS config
├── tsconfig.json        # TypeScript config
├── .gitignore           # Git ignore rules
├── README.md            # Project documentation
└── vercel.json          # Deployment config
```

Deployment Strategy

Environment Variables Needed

- `DATABASE_URL` - PostgreSQL connection
- `OPENAI_API_KEY` - For AI analysis
- `NEXTAUTH_URL` - Base URL
- `NEXTAUTH_SECRET` - Auth secret
- `EMAIL_SERVER_USER` - SMTP user
- `EMAIL_SERVER_PASSWORD` - SMTP password

Deployment Targets

1. **Development:** Local development server
2. **Staging:** Vercel preview deployments
3. **Production:** Vercel production deployment

Database Strategy

- **Development:** Local PostgreSQL or Supabase
- **Production:** Supabase or Railway PostgreSQL
- **Migrations:** Prisma handles schema changes

Module Development Plan

✅ Module 1: Framework (Current)

- Project structure setup
- Dependencies and configuration
- Deployment configuration

📄 Module 2: Frontend Foundation (Next)

- Basic layout and navigation
- Article display components
- Responsive design with Tailwind

📄 Module 3: Database & Models

- Prisma schema setup

- Database migrations
- CRUD operations

Module 4: News Automation

- RSS feed parsing
- Article scraping
- Content processing

Module 5: AI Analysis

- OpenAI integration
- Analysis generation
- Content comparison logic

Module 6: Email & Monetization

- Newsletter signup
- Email automation
- Ad integration setup

Key Architecture Decisions

1. **Next.js App Router** - Modern, SEO-friendly, deploys everywhere
2. **Prisma ORM** - Type-safe database access, handles migrations
3. **TypeScript** - Prevents deployment-breaking type errors
4. **Tailwind CSS** - Consistent styling, small bundle size
5. **Modular API Routes** - Scalable backend architecture
6. **Environment-based Config** - Seamless dev/staging/prod deployment