News Marketplace Website - System Architecture Report

Table of Contents

1. Project Overview

1.1 Project Description

The News Marketplace Website is a comprehensive digital platform designed to revolutionize the news and media industry by connecting content creators, publishers, and agencies in a unified ecosystem. The platform serves as a multi-faceted solution that combines content management, AI-assisted writing, publication marketplace, and agency directory functionalities.

1.2 Project Objectives

Primary Objectives:

- Create a centralized marketplace for news publications and media outlets
- Provide AI-assisted content creation tools for journalists and contributors
- Establish a comprehensive directory of news agencies and publications
- Streamline the content submission and review process
- Enable efficient communication between content creators and publishers
- Implement robust content protection and security measures

Secondary Objectives:

- Generate revenue through subscription models and premium features
- Build a community of verified journalists, editors, and contributors
- Provide analytics and insights for content performance
- Support multiple languages and international markets
- Establish partnerships with media organizations and agencies

1.3 Project Scope

In Scope:

- Multi-role user management system (Super Admin, Content Manager/Editor, Registered User, Agency)
- Content submission and management system with comprehensive metadata
- AI-assisted article writing using OpenAI and Gemini APIs
- Publication directory with detailed information and categorization
- Agency registration and management system
- Internal review and approval workflow for submitted content
- Automated email notifications and communication system
- Content protection mechanisms (disable right-click, copy/paste, screenshots)
- Multi-factor authentication system (email, SMS, WhatsApp)
- Responsive web application with mobile-first design
- Integration with third-party services (Firebase/Cloudinary, payment gateways)
- Comprehensive admin panel for system management
- Analytics and reporting capabilities

Out of Scope:

- Native mobile applications (iOS/Android)
- Video streaming or live broadcasting features
- Social media platform integration beyond basic sharing
- Real-time chat or messaging system
- Multi-language content translation services

- Third-party CMS integrations
- Advanced SEO tools beyond basic optimization

1.4 Project Success Criteria

- Successful deployment on Hostinger VPS with AlmaLinux
- User registration and authentication system with 99.9% uptime
- AI content generation with response time under 10 seconds
- Support for minimum 1,000 concurrent users
- Content submission and review workflow completion within defined SLAs
- Security compliance with industry standards
- Mobile responsiveness across all major devices and browsers
- Integration with all specified third-party services

2. Stakeholders

2.1 Primary Stakeholders

Project Sponsor/Owner

- **Role**: Overall project ownership and strategic direction
- **Responsibilities**: Project funding, high-level decision making, business strategy alignment
- **Interests**: ROI, market positioning, competitive advantage
- **Influence**: High
- **Engagement**: Monthly steering committee meetings

Product Manager

- **Role**: Product vision and roadmap management
- **Responsibilities**: Feature prioritization, user story creation, stakeholder communication
- **Interests**: User satisfaction, feature completeness, market fit
- **Influence**: High
- **Engagement**: Daily standups, weekly planning sessions

Development Team

- **Role**: Technical implementation and system development
- **Responsibilities**: Code development, testing, deployment, maintenance
- **Interests**: Technical excellence, code quality, development efficiency
- **Influence**: Medium-High
- **Engagement**: Daily development activities, sprint planning

2.2 Business Stakeholders

Content Creators (Journalists, Writers, Contributors)

- **Role**: Primary content providers and platform users
- **Responsibilities**: Content creation, submission, quality maintenance
- **Interests**: Easy content submission, AI assistance, fair compensation, platform visibility
- **Influence**: High (user adoption critical)
- **Engagement**: User feedback sessions, beta testing

Publishers and Media Outlets

- **Role**: Content consumers and platform partners
- **Responsibilities**: Content review, publication decisions, partnership agreements
- **Interests**: Quality content, efficient workflow, cost-effective sourcing
- **Influence**: High (revenue generation)
- **Engagement**: Partnership meetings, feedback collection

News Agencies

- **Role**: Institutional users and content distributors
- **Responsibilities**: Agency registration, content distribution, partnership management
- **Interests**: Platform integration, content reach, operational efficiency
- **Influence**: Medium-High
- **Engagement**: Regular partnership reviews

Content Managers/Editors

- **Role**: Content quality control and workflow management
- **Responsibilities**: Content review, approval/rejection, quality assurance
- **Interests**: Efficient review tools, quality metrics, workflow optimization
- **Influence**: Medium
- **Engagement**: Weekly workflow reviews

2.3 Technical Stakeholders

System Administrators

- **Role**: Infrastructure management and system maintenance
- **Responsibilities**: Server management, security, backups, monitoring
- **Interests**: System stability, security, performance optimization
- **Influence**: Medium
- **Engagement**: Daily monitoring, weekly maintenance windows

Database Administrators

- **Role**: Database design, optimization, and maintenance
- **Responsibilities**: Schema design, query optimization, data integrity
- **Interests**: Data consistency, performance, backup/recovery
- **Influence**: Medium
- **Engagement**: Database design reviews, performance monitoring

Security Team

- **Role**: Security architecture and compliance
- **Responsibilities**: Security requirements, vulnerability assessment, compliance
- **Interests**: Data protection, regulatory compliance, threat mitigation
- **Influence**: High (regulatory requirements)
- **Engagement**: Security reviews, compliance audits

2.4 External Stakeholders

AI Service Providers (OpenAI, Gemini)

- **Role**: AI content generation services
- **Responsibilities**: API reliability, service quality, cost management
- **Interests**: API usage growth, service stability
- **Influence**: Medium
- **Engagement**: Service level agreements, technical support

Cloud Service Providers (Firebase, Cloudinary)

- **Role**: Storage and media processing services
- **Responsibilities**: Service availability, data security, performance
- **Interests**: Service utilization, customer satisfaction
- **Influence**: Medium
- **Engagement**: Service monitoring, support tickets

Hosting Provider (Hostinger)

- **Role**: Infrastructure hosting and support
- **Responsibilities**: Server availability, network performance, technical support
- **Interests**: Customer retention, service quality
- **Influence**: Medium
- **Engagement**: Support tickets, service reviews

3. Functional Requirements

3.1 User Management and Authentication

FR-001: User Registration System

- The system shall support multi-role user registration (Super Admin, Content Manager/Editor, Registered User, Agency)
- The system shall implement three-factor authentication (email, SMS, WhatsApp)
- The system shall validate email addresses through verification links
- The system shall store user profile information including contact details and preferences
- The system shall support role-based access control with granular permissions

FR-002: Authentication and Session Management

- The system shall implement secure login with JWT token-based authentication
- The system shall enforce single-device login policy per user
- The system shall automatically logout users after 30 minutes of inactivity
- The system shall provide password reset functionality via email
- The system shall maintain session logs for security auditing

FR-003: User Profile Management

- Users shall be able to update their profile information
- Users shall be able to upload and manage profile images
- Users shall be able to set notification preferences
- Users shall be able to view their activity history
- Users shall be able to manage their subscription status

3.2 Content Management System

FR-004: Article Submission System

- Registered users shall be able to submit articles through a comprehensive form
- The system shall capture article metadata including title, category, tags, and description
- The system shall support rich text editing with formatting options
- The system shall allow multiple media file uploads (images, documents)
- The system shall validate article content against defined criteria

FR-005: Content Review Workflow

- The system shall queue submitted articles for internal review
- Content managers shall be able to approve or reject submitted articles
- The system shall provide feedback mechanisms for rejected content
- The system shall track article status throughout the review process
- The system shall maintain audit trails for all review decisions

FR-006: Publication Management

- The system shall maintain a comprehensive database of publications
- The system shall store publication details including pricing, TAT, and requirements
- The system shall categorize publications by industry, region, and content type
- The system shall track publication performance metrics (DA, DR)
- The system shall support publication search and filtering capabilities

3.3 AI-Assisted Content Creation

FR-007: AI Content Generation

- The system shall integrate with OpenAI and Gemini APIs for content generation
- Users shall be able to generate articles based on guided questions
- The system shall provide real-time content suggestions and improvements
- The system shall allow users to review and edit AI-generated content
- The system shall track AI usage and associated costs per user

FR-008: Content Enhancement Tools

- The system shall provide grammar and style checking capabilities
- The system shall suggest SEO optimizations for article content
- The system shall offer content structure recommendations
- The system shall provide keyword suggestions based on article topic
- The system shall support multiple content tones and styles

3.4 Communication and Notification System

FR-009: Email Integration

- The system shall send automated confirmation emails for article submissions
- The system shall notify publishers of new content availability
- The system shall send status updates to content creators
- The system shall provide email templates for different notification types
- The system shall track email delivery and engagement metrics

FR-010: Multi-Channel Notifications

- The system shall support SMS notifications for critical updates
- The system shall integrate with WhatsApp for communication
- The system shall provide in-app notifications for user activities
- The system shall allow users to configure notification preferences
- The system shall maintain notification history and delivery status

3.5 Agency and Partnership Management

FR-011: Agency Registration

- News agencies shall be able to register and create detailed profiles
- The system shall support agency verification and approval process
- Agencies shall be able to manage their publication listings
- The system shall provide agency dashboard for content and partnership management
- The system shall support agency-specific branding and customization

FR-012: Partnership Management

- The system shall facilitate media partnerships for events and collaborations
- The system shall provide partnership agreement templates and management
- The system shall track partnership performance and outcomes
- The system shall support revenue sharing and commission tracking
- The system shall provide partnership analytics and reporting

3.6 Search and Discovery

FR-013: Content Search System

- Users shall be able to search articles by keywords, categories, and tags
- The system shall provide advanced search filters and sorting options
- The system shall implement full-text search capabilities
- The system shall provide search suggestions and auto-complete
- The system shall track search queries and popular terms

FR-014: Content Categorization

- The system shall support hierarchical content categorization
- The system shall provide category-based content browsing
- The system shall allow dynamic category creation and management
- The system shall support content tagging and tag-based discovery
- The system shall provide trending content identification

3.7 Analytics and Reporting

FR-015: User Analytics

- The system shall track user engagement and activity metrics
- The system shall provide user behavior analytics and insights
- The system shall generate user performance reports

- The system shall track content creation and submission statistics
- The system shall provide user retention and churn analysis

FR-016: Content Analytics

- The system shall track article performance metrics (views, engagement)
- The system shall provide content quality and approval rate analytics
- The system shall generate publisher and agency performance reports
- The system shall track AI usage and cost analytics
- The system shall provide revenue and monetization reports

3.8 Administrative Functions

FR-017: System Administration

- Super admins shall have full system access and control
- The system shall provide comprehensive admin dashboard
- Admins shall be able to manage user accounts and permissions
- The system shall support system configuration and settings management
- The system shall provide system health monitoring and alerts

FR-018: Content Moderation

- The system shall provide content moderation tools and workflows
- Moderators shall be able to flag and review inappropriate content
- The system shall support automated content filtering and detection
- The system shall maintain moderation logs and decision history
- The system shall provide appeals process for moderation decisions

4. Non-functional Requirements

4.1 Performance Requirements

NFR-001: Response Time

- Web pages shall load within 3 seconds under normal load conditions
- AI content generation shall complete within 10 seconds
- Database queries shall execute within 2 seconds for 95% of requests
- File uploads shall support up to 10MB files with progress indicators
- Search results shall be returned within 1 second

NFR-002: Throughput

- The system shall support minimum 1,000 concurrent users
- The system shall handle 100 article submissions per hour
- The system shall process 500 search queries per minute
- The system shall support 50 simultaneous AI content generation requests
- The system shall handle 10,000 page views per hour

NFR-003: Scalability

- The system architecture shall support horizontal scaling
- Database shall support read replicas for improved performance
- The system shall handle 10x current load with infrastructure scaling
- CDN integration shall be supported for global content delivery
- Load balancing shall be implemented for high availability

4.2 Security Requirements

NFR-004: Authentication and Authorization

- All user passwords shall be encrypted using industry-standard hashing
- JWT tokens shall expire within 24 hours with refresh token mechanism
- Multi-factor authentication shall be enforced for admin users
- Role-based access control shall be implemented throughout the system
- Session management shall prevent concurrent logins from multiple devices

NFR-005: Data Protection

- All data transmission shall be encrypted using HTTPS/TLS 1.3
- Sensitive data shall be encrypted at rest in the database
- Personal data shall comply with GDPR and privacy regulations
- Data backup shall be encrypted and stored securely
- API endpoints shall implement rate limiting and DDoS protection

NFR-006: Content Protection

- Right-click functionality shall be disabled on protected content
- Screenshot and copy/paste functionality shall be restricted
- Content watermarking shall be implemented for premium articles
- Access logging shall track all content access attempts
- Digital rights management shall be enforced for premium content

4.3 Usability Requirements

NFR-007: User Interface

- The interface shall be intuitive and require minimal training
- The system shall provide consistent navigation across all pages
- Error messages shall be clear and actionable
- The system shall provide contextual help and documentation
- Accessibility standards (WCAG 2.1 AA) shall be met

NFR-008: Mobile Responsiveness

- The system shall be fully functional on mobile devices
- Touch interfaces shall be optimized for mobile interaction
- Mobile page load times shall be under 3 seconds
- Offline functionality shall be provided for basic operations
- Progressive Web App (PWA) features shall be implemented

4.4 Reliability Requirements

NFR-009: Availability

- The system shall maintain 99.9% uptime (maximum 8.76 hours downtime per year)
- Planned maintenance windows shall not exceed 4 hours monthly
- System recovery time shall be less than 15 minutes for critical failures
- Database backup and recovery procedures shall be automated
- Disaster recovery plan shall ensure 24-hour maximum recovery time

NFR-010: Error Handling

- The system shall gracefully handle all error conditions
- User-friendly error messages shall be displayed for all failures
- System errors shall be logged with sufficient detail for debugging
- Automatic retry mechanisms shall be implemented for transient failures
- Fallback mechanisms shall be provided for external service failures

4.5 Compatibility Requirements

NFR-011: Browser Compatibility

- The system shall support Chrome, Firefox, Safari, and Edge browsers
- Browser versions from the last 2 years shall be supported
- JavaScript shall be required with graceful degradation where possible
- Cross-browser testing shall be performed for all major features
- Browser-specific optimizations shall be implemented where necessary

NFR-012: Platform Compatibility

- The system shall run on AlmaLinux operating system
- PostgreSQL database compatibility shall be maintained
- Node.js runtime compatibility shall be ensured
- Third-party service APIs shall be abstracted for easy switching
- Container deployment (Docker) shall be supported

4.6 Maintainability Requirements

NFR-013: Code Quality

- Code shall follow established coding standards and best practices
- Code coverage shall be maintained at minimum 80%
- Automated testing shall be implemented for all critical functions
- Code documentation shall be comprehensive and up-to-date
- Version control and branching strategies shall be enforced

NFR-014: Monitoring and Logging

- Comprehensive application logging shall be implemented
- Performance monitoring and alerting shall be configured
- User activity tracking shall be implemented for analytics
- System health dashboards shall be provided
- Log retention policies shall be enforced for compliance

4.7 Compliance Requirements

NFR-015: Regulatory Compliance

- GDPR compliance shall be maintained for European users
- Data retention policies shall be implemented and enforced
- User consent management shall be provided
- Data export and deletion capabilities shall be available
- Privacy policy and terms of service shall be maintained

NFR-016: Industry Standards

- Web Content Accessibility Guidelines (WCAG) 2.1 AA compliance
- OWASP security guidelines shall be followed
- RESTful API design principles shall be implemented
- JSON data format standards shall be used
- HTTP status codes shall be used appropriately

5. Implementation Approach

We will implement a comprehensive News Marketplace Website with the following key features:

Phase 1: Core Platform Development

1. **User Management System** - Multi-role authentication (Writers, Readers, Admins)
2. **Content Management** - Article creation, editing, and publishing workflow
3. **AI Integration** - OpenAI/Gemini-powered content assistance
4. **Basic Search & Discovery** - Article browsing and categorization

Phase 2: Advanced Features

1. **Payment Integration** - Subscription management and individual article purchases
2. **Social Features** - Comments, likes, bookmarks, and sharing
3. **Advanced Analytics** - User engagement and content performance metrics
4. **Admin Panel** - Content moderation and system management

Phase 3: Optimization & Scaling

1. **Performance Optimization** - Caching, CDN integration, and database optimization
2. **Mobile Responsiveness** - Enhanced mobile experience
3. **SEO Enhancement** - Advanced SEO features and social media integration
4. **Security Hardening** - Advanced security measures and monitoring

Critical Requirements Addressed:

- **Scalable Architecture**: Microservices-ready design with clear separation of concerns
- **AI-Powered Content Creation**: Seamless integration with OpenAI and Gemini APIs
- **Multi-Role User System**: Flexible role-based access control
- **Payment Processing**: Secure subscription and transaction management
- **Content Workflow**: Comprehensive article lifecycle from creation to publication
- **Real-time Features**: Live notifications and real-time updates

6. Technology Stack

Frontend Technologies

- **Framework**: React.js 18+ with TypeScript
- **Styling**: TailwindCSS for utility-first CSS
- **State Management**: React Context API + useReducer for complex state
- **Routing**: React Router v6 for client-side navigation

- **HTTP Client**: Axios for API communication
- **Form Handling**: React Hook Form with Yup validation
- **Rich Text Editor**: Draft.js or Quill.js for article editing
- **Testing**: Jest + React Testing Library + Playwright for E2E testing

Backend Technologies

- **Runtime**: Node.js 18+ LTS
- **Framework**: Express.js with TypeScript
- **Database**: PostgreSQL 15+ for primary data storage
- **Caching**: Redis for session management and API caching
- **ORM**: Prisma or Sequelize for database operations
- **Authentication**: JWT tokens with refresh token rotation
- **File Upload**: Multer with cloud storage integration
- **API Documentation**: Swagger/OpenAPI 3.0

AI & External Services

- **AI Engines**: OpenAI GPT-4 and Google Gemini Pro APIs
- **File Storage**: Firebase Storage and Cloudinary for media processing
- **Payment Processing**: Stripe for subscriptions and one-time payments
- **Email Service**: SendGrid or Mailgun for transactional emails
- **Search Engine**: PostgreSQL full-text search with potential Elasticsearch upgrade

Infrastructure & Deployment

- **Hosting**: Hostinger VPS with AlmaLinux 9
- **Web Server**: Nginx as reverse proxy and static file server
- **SSL**: Let's Encrypt for HTTPS certificates
- **Process Management**: PM2 for Node.js application management
- **Monitoring**: Custom logging with Winston + optional Grafana/Prometheus
- **Backup**: Automated PostgreSQL backups with retention policies

7. User & UI Interaction Patterns

1. Guest User Interactions

- **Landing Page Access**: Browse featured articles and categories without authentication
- **Content Discovery**: Search and filter articles by category, keywords, and popularity
- **Article Preview**: Read article excerpts and view basic metadata

- **Registration Flow**: Simple sign-up process with email verification
- **Social Sharing**: Share articles on social media platforms

2. Registered Reader Interactions

- **Personalized Dashboard**: View recommended articles based on reading history
- **Full Article Access**: Read complete articles based on subscription level
- **Social Engagement**: Like, comment, and bookmark articles
- **Subscription Management**: Upgrade/downgrade subscription plans
- **Profile Customization**: Manage personal information and preferences
- **Payment History**: View transaction history and download invoices

3. Content Writer Interactions

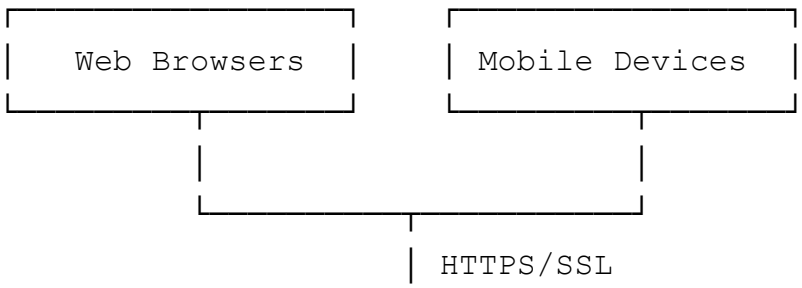
- **Writer Dashboard**: Access writing tools and article management interface
- **AI-Assisted Writing**: Use AI suggestions for content generation and improvement
- **Article Editor**: Rich text editing with media upload capabilities
- **Draft Management**: Save, edit, and organize article drafts
- **Publication Workflow**: Submit articles for review and track approval status
- **Performance Analytics**: View article engagement metrics and reader feedback

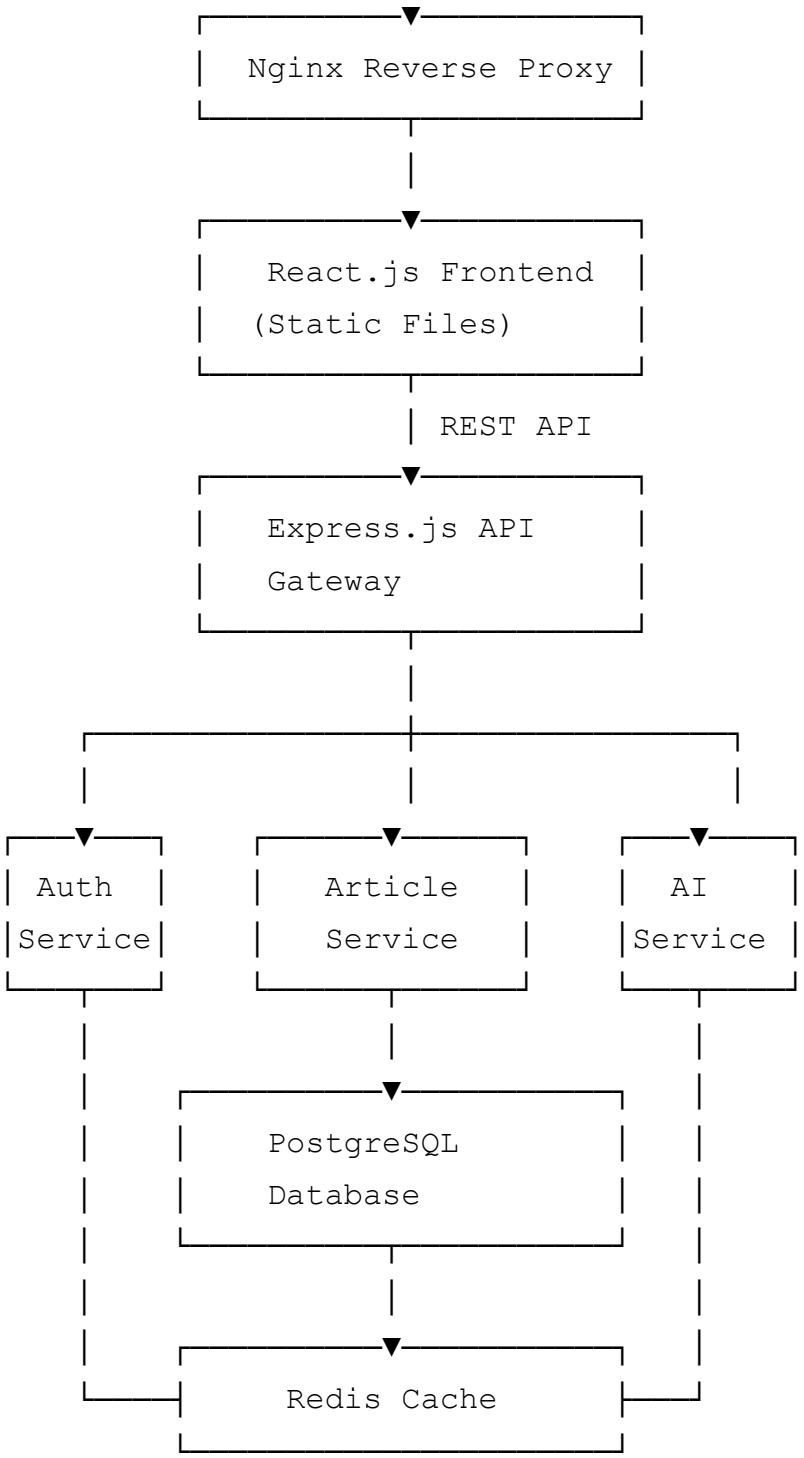
4. Administrator Interactions

- **Admin Dashboard**: Comprehensive system overview with key metrics
- **User Management**: Monitor user activities and manage account statuses
- **Content Moderation**: Review, approve, or reject submitted articles
- **System Configuration**: Manage categories, subscription plans, and system settings
- **Analytics & Reporting**: Generate detailed reports on platform performance
- **Security Monitoring**: Track suspicious activities and manage security settings

8. System Architecture

The system follows a layered architecture pattern with clear separation between presentation, business logic, and data layers:





Key Architectural Principles:

- **Modularity**: Each service handles specific business domains
- **Scalability**: Horizontal scaling capability for individual services
- **Security**: JWT-based authentication with role-based access control
- **Performance**: Multi-layer caching strategy with Redis and CDN
- **Reliability**: Error handling, logging, and graceful degradation

9. Data Structures & Interfaces

Core Data Models

```
// User Management
interface User {
  id: string;
  email: string;
  passwordHash: string;
  role: 'writer' | 'reader' | 'admin';
  firstName: string;
  lastName: string;
  profileImage?: string;
  bio?: string;
  isVerified: boolean;
  isActive: boolean;
  createdAt: Date;
  updatedAt: Date;
}

// Content Management
interface Article {
  id: string;
  title: string;
  slug: string;
  content: string;
  excerpt: string;
  featuredImage?: string;
  status: 'draft' | 'pending' | 'published' | 'rejected';
  viewCount: number;
  likeCount: number;
  isPremium: boolean;
  price?: number;
  seoTitle?: string;
  seoDescription?: string;
  keywords: string[];
  publishedAt?: Date;
  authorId: string;
  categoryId: string;
  approvedBy?: string;
  createdAt: Date;
  updatedAt: Date;
}

// AI Integration
interface AIGeneration {
  id: string;
  prompt: string;
```

```
  generatedContent: string;
  modelUsed: 'openai' | 'gemini';
  tokensUsed: number;
  cost: number;
  userId: string;
  articleId?: string;
  createdAt: Date;
}

// Payment System
interface Subscription {
  id: string;
  userId: string;
  planType: 'basic' | 'premium' | 'enterprise';
  status: 'active' | 'cancelled' | 'expired';
  startDate: Date;
  endDate: Date;
  autoRenew: boolean;
  createdAt: Date;
}
```

API Interfaces

```
// Authentication APIs
interface AuthAPI {
  register(userData: RegisterRequest): Promise<AuthResponse>;
  login(credentials: LoginRequest): Promise<AuthResponse>;
  refreshToken(token: string): Promise<TokenResponse>;
  logout(token: string): Promise<void>;
  forgotPassword(email: string): Promise<void>;
  resetPassword(token: string, newPassword: string): Promise<void>;
}

// Article Management APIs
interface ArticleAPI {
  createArticle(articleData: CreateArticleRequest): Promise<Article>;
  updateArticle(id: string, updates: UpdateArticleRequest): Promise<Artic
  publishArticle(id: string): Promise<Article>;
  deleteArticle(id: string): Promise<void>;
  getArticle(id: string): Promise<Article>;
  searchArticles(query: SearchRequest): Promise<PaginatedResponse<Article
  getArticlesByCategory(categoryId: string): Promise<PaginatedResponse<Ar
```

```
  }

  // AI Service APIs
  interface AIAPI {
    generateContent(prompt: GenerateContentRequest): Promise<AIResponse>;
    getSuggestions(context: string): Promise<string[]>;
    improveContent(content: string, instructions: string): Promise<string>;
  }
```

10. Program Call Flow

Article Creation & Publishing Flow

```
User → Frontend → API Gateway → Auth Middleware → Article Service → AI
Service → Database
  |        |           |              |               |              |
|
  |        |           |              |               |              |
|
  ▼        ▼           ▼              ▼               ▼              ▼
▼
Login → Dashboard → Create Article → Validate JWT → Process Request →
Generate AI → Store Data
  |        |           |              |               |           Content
|
  |        |           |              |               |              |
|
  ▼        ▼           ▼              ▼               ▼              ▼
▼
Success → UI Update → Show Editor → Grant Access → Save Draft → Return
Content → Confirm Save
```

Detailed Sequence:

1. **User Authentication**: JWT validation and role verification
2. **Article Creation**: Initialize editor with AI assistance options
3. **Content Generation**: AI API calls for content suggestions
4. **Media Upload**: File processing and cloud storage integration
5. **Draft Management**: Periodic auto-save and version control
6. **Review Process**: Admin approval workflow for content quality
7. **Publication**: SEO optimization and search index updates

8. **Notification**: Real-time updates to subscribers and followers

11. Database Design

Entity Relationship Overview

The database design follows a normalized approach with clear relationships:

- **Users** have many **Articles**, **Subscriptions**, and **Payments**
- **Articles** belong to **Categories** and have many **Comments**, **Likes**, and **Bookmarks**
- **Subscriptions** are linked to **Payments** for billing history
- **AI Generations** track usage for cost management and analytics

Key Database Features

1. **Indexing Strategy**: Optimized indexes for search queries and performance
2. **Constraints**: Foreign key relationships and data integrity rules
3. **Triggers**: Automated updates for denormalized fields (like counts)
4. **Partitioning**: Time-based partitioning for large tables (analytics data)
5. **Backup Strategy**: Daily incremental and weekly full backups

12. UI Navigation Flow

The navigation follows a hierarchical structure with maximum 3-4 levels depth:

```
Landing Page
├── Authentication (Login/Register)
├── Content Discovery
│   ├── Browse Articles
│   ├── Search Results
│   └── Category Views
│       └── Article Detail
├── User Dashboard
│   ├── Writer Dashboard
│   │   ├── Article Management
│   │   │   ├── Create/Edit Article
│   │   │   └── AI Assistant
│   │   └── Analytics
│   ├── Reader Dashboard
│   │   ├── Bookmarks
```

```
|   |   └── Reading History
|   └── Profile Settings
└── Admin Panel (Admin Only)
    ├── User Management
    ├── Content Moderation
    └── System Analytics
```

Navigation Principles:

- **Breadcrumb Navigation**: Clear path indication at all levels
- **Quick Actions**: Frequently used functions accessible from any page
- **Responsive Design**: Consistent experience across all device sizes
- **Search Integration**: Global search available from every page

13. File & Folder Structure

The project follows a modular structure with clear separation of concerns:

Frontend Structure (React.js)

- **Components**: Reusable UI components organized by feature
- **Pages**: Route-level components for different application views
- **Services**: API communication and external service integrations
- **Hooks**: Custom React hooks for shared logic
- **Context**: Global state management for authentication and themes
- **Utils**: Helper functions, validators, and constants

Backend Structure (Node.js)

- **Controllers**: Request handling and response formatting
- **Services**: Business logic and external API integrations
- **Models**: Database schema definitions and relationships
- **Middleware**: Authentication, validation, and error handling
- **Routes**: API endpoint definitions and routing logic
- **Utils**: Shared utilities, database connections, and configurations

14. CRUD Operations Mapping

Comprehensive mapping of all Create, Read, Update, Delete operations:

- **User Management**: Full CRUD with role-based access control

- **Article Management**: Complete lifecycle from draft to publication
- **Payment Processing**: Secure transaction handling with audit trails
- **Content Interaction**: Social features with real-time updates
- **Admin Operations**: System management and content moderation

15. Additional System Diagrams

Use Case Diagram

Illustrates all user interactions and system boundaries for different user roles.

Data Flow Diagrams (Level 0 & 1)

Shows information flow through the system at different abstraction levels.

Component Diagram

Details the internal structure and dependencies between system components.

Deployment Diagram

Visualizes the physical deployment of software components on hardware infrastructure.

16. Deployment Architecture

Production Environment Setup

Server Configuration:

- **OS**: AlmaLinux 9 on Hostinger VPS
- **Web Server**: Nginx with SSL termination
- **Application**: Node.js with PM2 process management
- **Database**: PostgreSQL with connection pooling
- **Caching**: Redis for session and API caching

Security Measures:

- **SSL/TLS**: Let's Encrypt certificates with auto-renewal
- **Firewall**: UFW configuration with minimal open ports
- **Authentication**: JWT with refresh token rotation

- **Rate Limiting**: API endpoint protection against abuse
- **Input Validation**: Comprehensive data sanitization

Monitoring & Logging:

- **Application Logs**: Winston with log rotation
- **Access Logs**: Nginx access and error logging
- **Performance Monitoring**: Custom metrics and alerts
- **Backup Strategy**: Automated database and file backups

17. Security Considerations

Authentication & Authorization

- **Multi-factor Authentication**: Optional 2FA for enhanced security
- **Password Policies**: Strong password requirements with hashing
- **Session Management**: Secure JWT implementation with blacklisting
- **Role-based Access**: Granular permissions for different user types

Data Protection

- **Encryption**: At-rest and in-transit data encryption
- **Input Sanitization**: XSS and SQL injection prevention
- **API Security**: Rate limiting and request validation
- **File Upload Security**: Malware scanning and type validation

Privacy Compliance

- **GDPR Compliance**: User data rights and consent management
- **Data Retention**: Automated cleanup of expired data
- **Audit Trails**: Comprehensive logging of sensitive operations
- **Privacy Controls**: User data export and deletion capabilities

18. Performance Optimization

Caching Strategy

- **Redis Caching**: API responses and session data
- **Browser Caching**: Static assets with appropriate headers
- **CDN Integration**: Global content delivery for media files

- **Database Caching**: Query result caching for frequent operations

Database Optimization

- **Index Optimization**: Strategic indexing for query performance
- **Connection Pooling**: Efficient database connection management
- **Query Optimization**: Efficient SQL queries with proper joins
- **Partitioning**: Large table partitioning for better performance

Frontend Optimization

- **Code Splitting**: Lazy loading of components and routes
- **Image Optimization**: Responsive images with multiple formats
- **Bundle Optimization**: Tree shaking and minification
- **Progressive Loading**: Skeleton screens and progressive enhancement

19. Unclear Aspects & Assumptions

Clarifications Needed

1. **AI Usage Limits**: What are the monthly token limits and cost constraints for AI services?
2. **Payment Integration**: Which specific payment gateways are preferred (Stripe, PayPal, others)?
3. **Content Moderation**: What are the specific content guidelines and moderation policies?
4. **Subscription Tiers**: What are the exact features and pricing for different subscription levels?
5. **Mobile App**: Is a native mobile application planned for future development?

Assumptions Made

1. **User Base**: Assuming initial user base of 1,000-10,000 users with potential for growth
2. **Content Volume**: Expecting 100-1,000 articles per month initially
3. **Geographic Scope**: Assuming English-language content with potential for internationalization
4. **Compliance**: Assuming GDPR compliance requirements for European users
5. **Scalability**: Designing for horizontal scaling when user base exceeds 10,000 users

Technical Decisions

1. **Database Choice**: PostgreSQL chosen for ACID compliance and advanced features
2. **AI Integration**: Dual AI provider setup for redundancy and feature comparison
3. **Authentication**: JWT chosen for stateless authentication and scalability
4. **File Storage**: Cloud storage preferred over local storage for scalability

5. **Caching**: Redis chosen for its advanced data structures and pub/sub capabilities